

Fig. 1A (Prior Art)

2/73

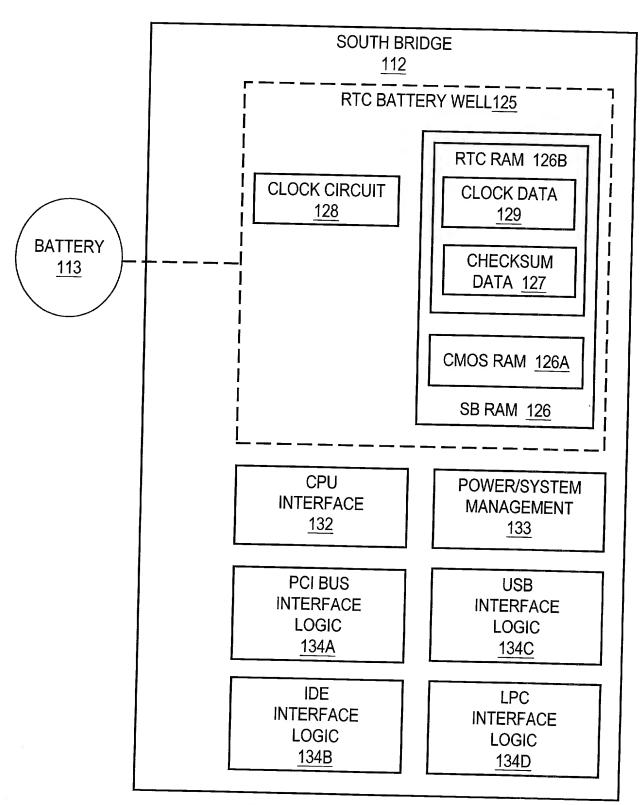


Fig. 1B (Prior Art)

POWER SUPPLY INITIALIZATION POWER SUPPLY GENERATES A POWER GOOD SIGNAL TO THE NORTH BRIDGE 136

UP RECEIVING THE POWER GOOD SIGNAL, THE SOUTH BRIDGE STOPS ASSERTING THE RESET SIGNAL FOR THE PROCESSOR 138

THE PROCESSOR READS THE DEFAULT JUMP LOCATION, USUALLY AT FFFF0h 140

THE PROCESSOR JUMPS TO THE BIOS CODE LOCATION IN THE ROM BIOS, COPIES THE BIOS CODE TO RAM, AND BEGINS PROCESSING BIOS CODE INSTRUCTIONS FROM RAM 142

BIOS CODE PERFORMS POWER ON SELF TEST (POST) 144

BIOS CODE LOOKS FOR ADDITIONAL BIOS CODE, SUCH AS VIDEO @ C000h AND ATA/IDE HARD DRIVE BIOS CODE @ C800h, AND DISPLAYS A START-UP INFORMATION SCREEN 146

BIOS CODE PERFORMS ADDITIONAL SYSTEM TESTS, SUCH AS THE RAM COUNT-UP TEST, AND SYSTEM INVENTORY, SUCH AS IDENTIFYING COM AND LPT PORTS 148

BIOS CODE IDENTIFIES PLUG-N-PLAY AND OTHER SIMILAR DEVICES AND DISPLAYS A SUMMARY SCREEN 150

BIOS CODE IDENTIFIES THE BOOT LOCATION 152

BIOS CODE CALLS THE BOOT SECTOR CODE TO BOOT THE COMPUTER SYSTEM 154

Fig. 2A (Prior Art)

*-* 170 INTERRUPT CONTROLLER RECEIVES A REQUEST FOR SYSTEM MANAGEMENT MODE (SMM) 172 INTERRUPT CONTROLLER SIGNALS THE REQUEST FOR SMM TO THE PROCESSOR BY ASSERTING THE SYSTEM MANAGEMENT INTERRUPT (SMI#) SIGNAL 174 PROCESSOR RECOGNIZES THE REQUEST FOR SMM AND ASSERTS THE SMI ACTIVE (SMIACT#) SIGNAL 176 SYSTEM RECOGINIZES THE SMIACT# SIGNAL, DISABLES ACCESS TO RAM. AND ENABLES ACCESS TO SYSTEM MANAGEMENT RAM (SMRAM) SPACE 178 CURRENT PROCESSOR STATE IS SAVED TO SMRAM 180 PROCESSOR RESETS TO SMM DEFAULT STATE AND ENTERS SMM 182 PROCESSOR READS DEFAULT POINTER AND JUMPS INTO SMRAM SPACE 184 STATUS REGISTERS ARE CHECKED TO IDENTIFY THE SMI REQUEST 186 SMI HANDLER SERVICES THE SMI REQUEST 188 SMI HANDLER ISSUES RETURN FROM SMM (RSM) INSTRUCTION TO PROCESSOR 190 PROCESSOR RESTORES SAVED STATE INFORMATION AND CONTINUES

Fig. 2B (Prior Art)

NORMAL OPERATION 192

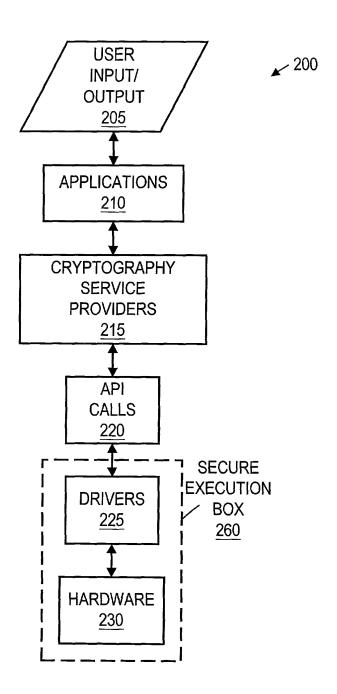
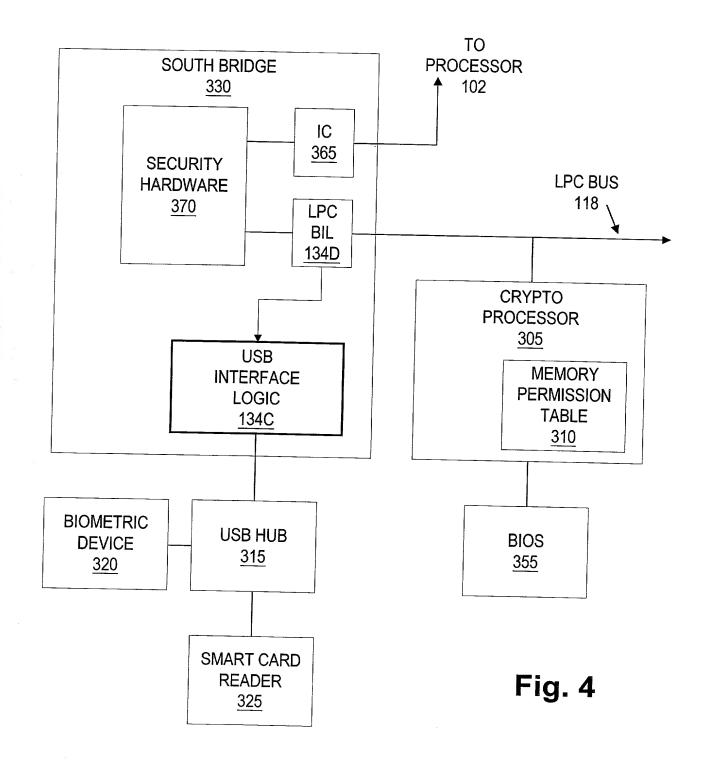


Fig. 3

6 / 73



7/73

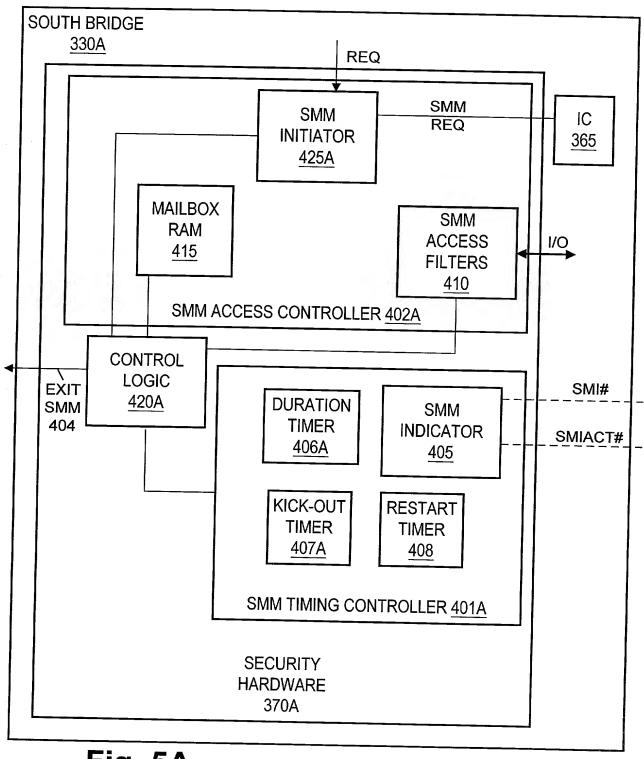


Fig. 5A

8 / 73

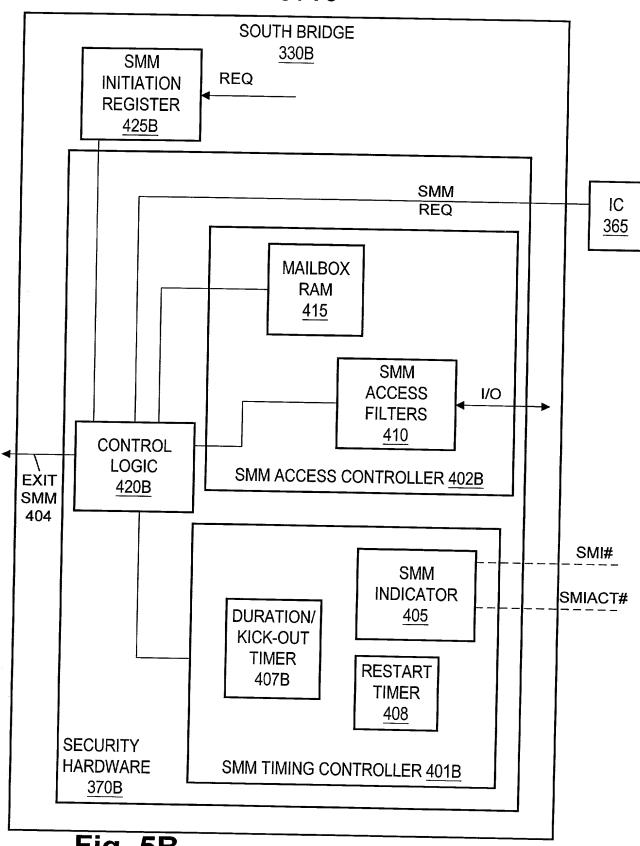


Fig. 5B

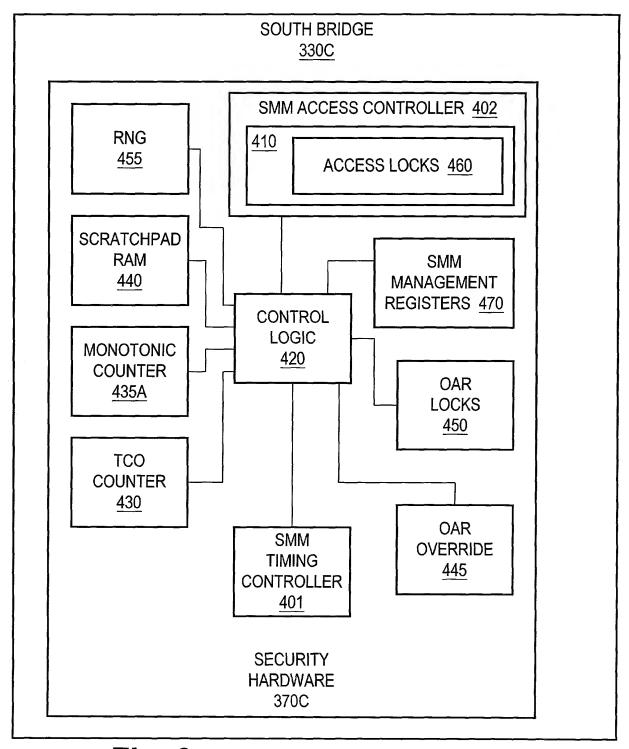
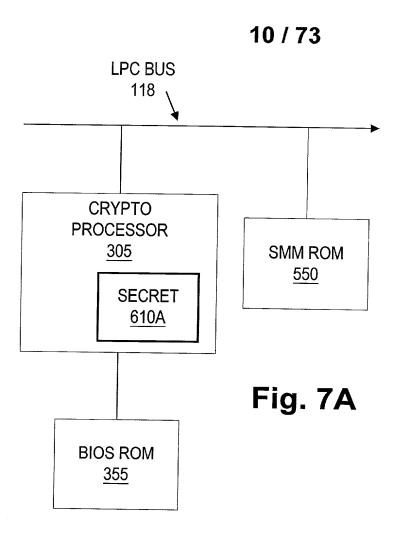


Fig. 6



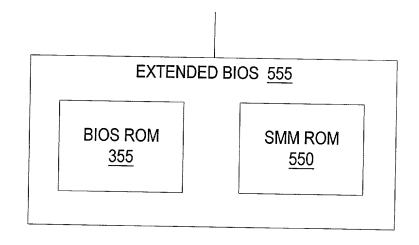


Fig. 7B

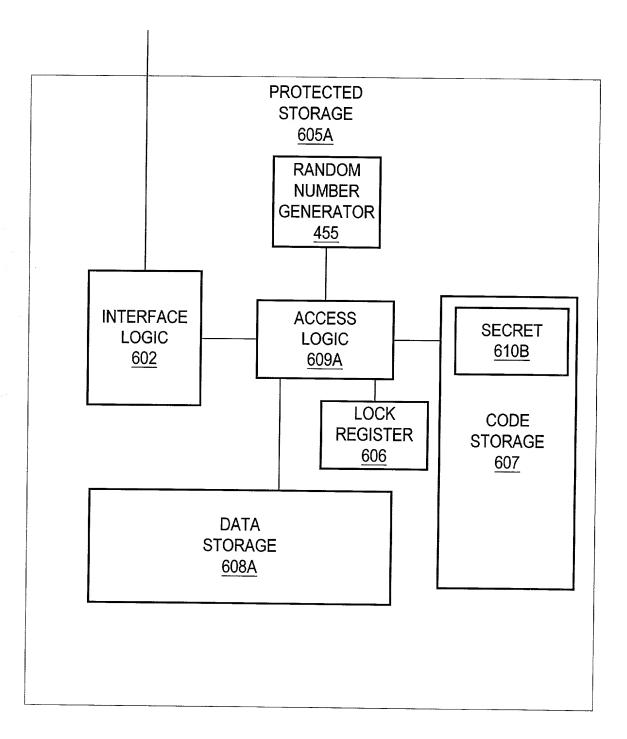


Fig. 7C

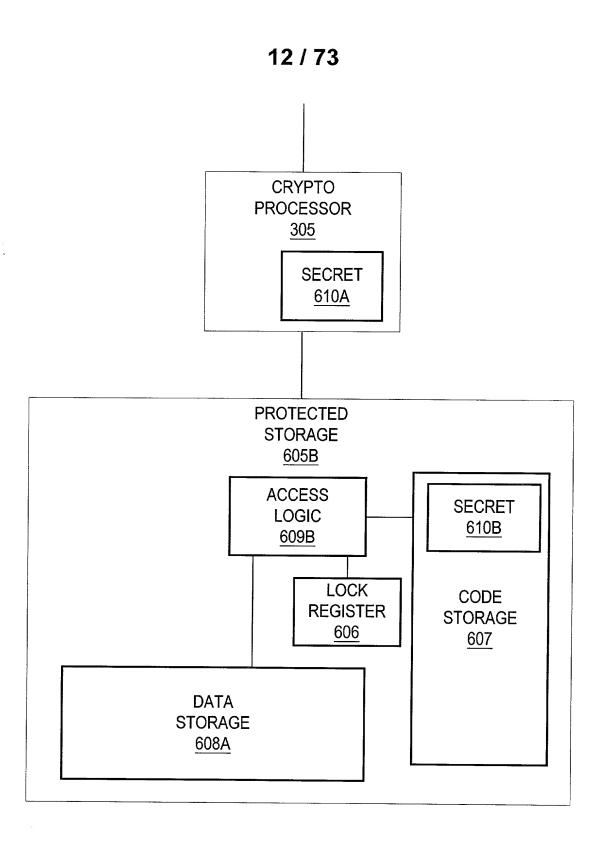


Fig. 7D

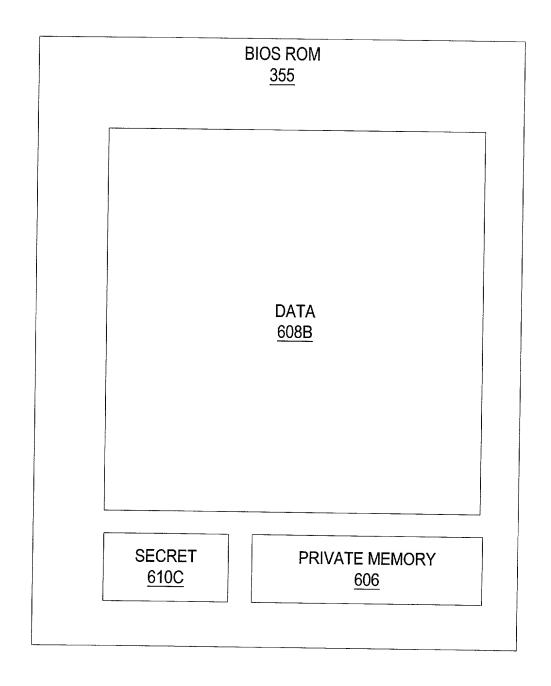


Fig. 8A

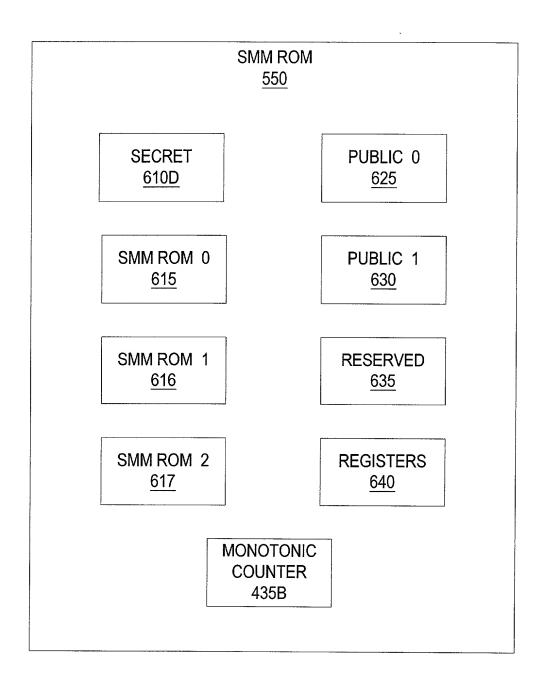


Fig. 8B

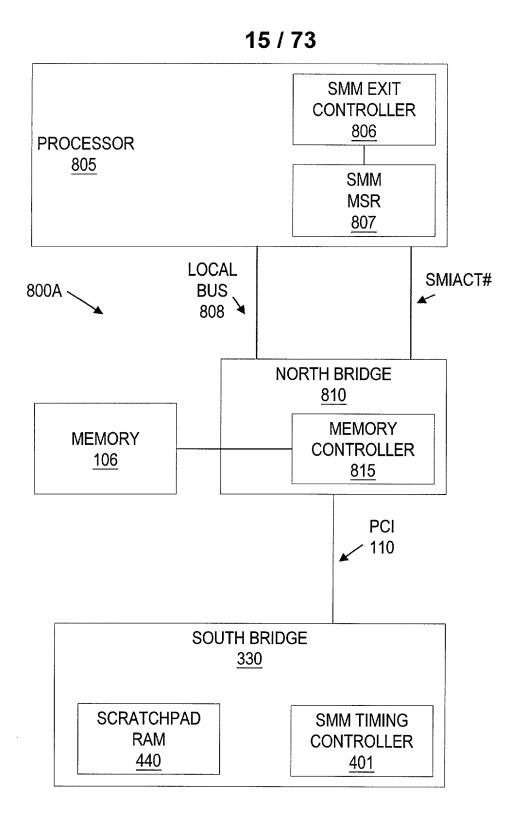


Fig. 9A

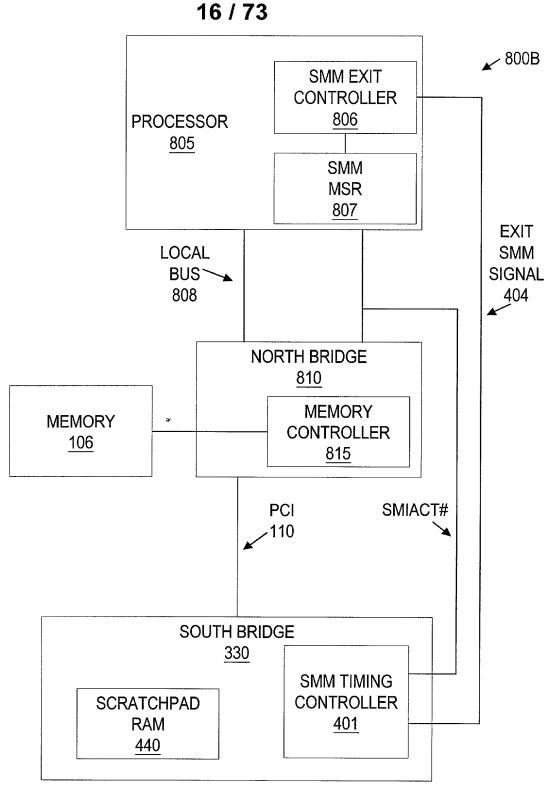
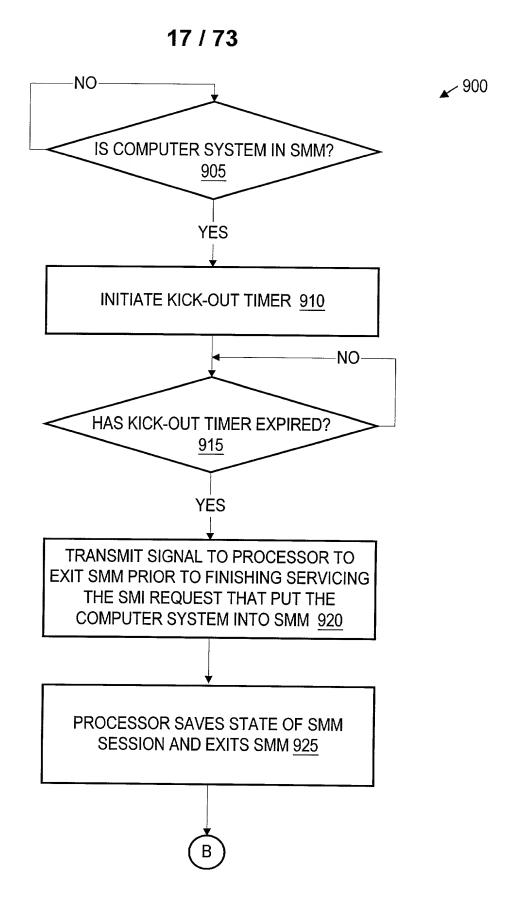


Fig. 9B



**Fig. 10A** 

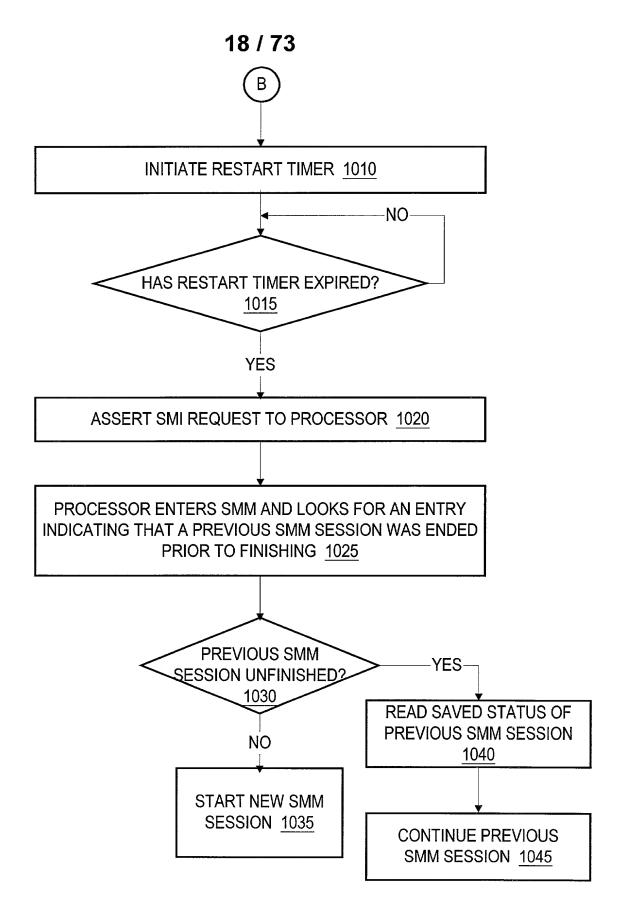
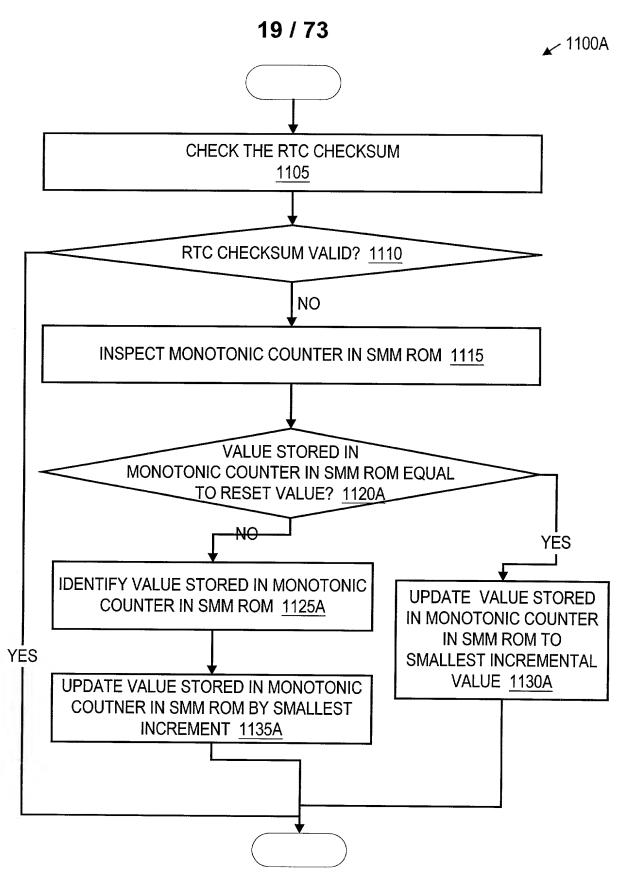


Fig. 10B



**Fig. 11A** 

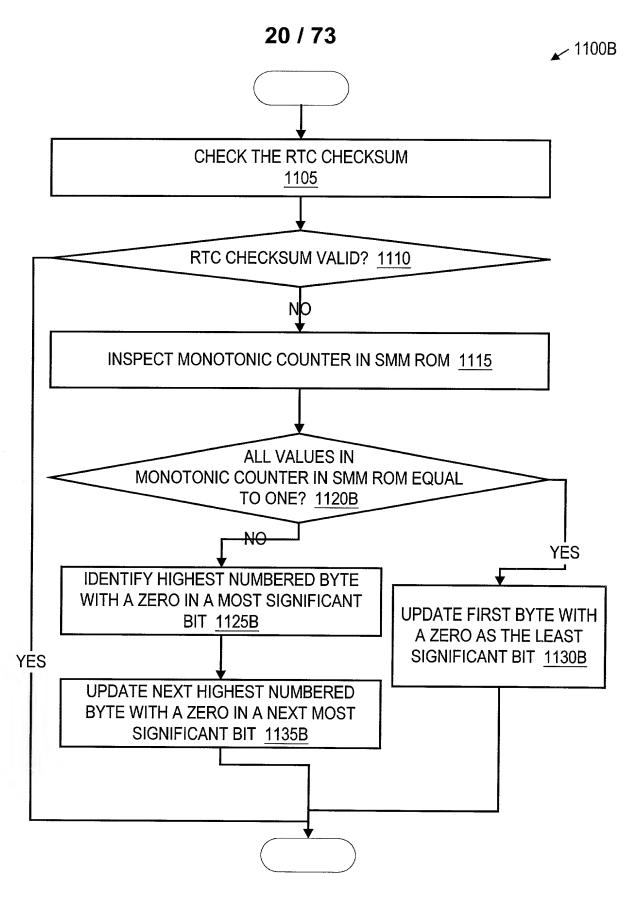
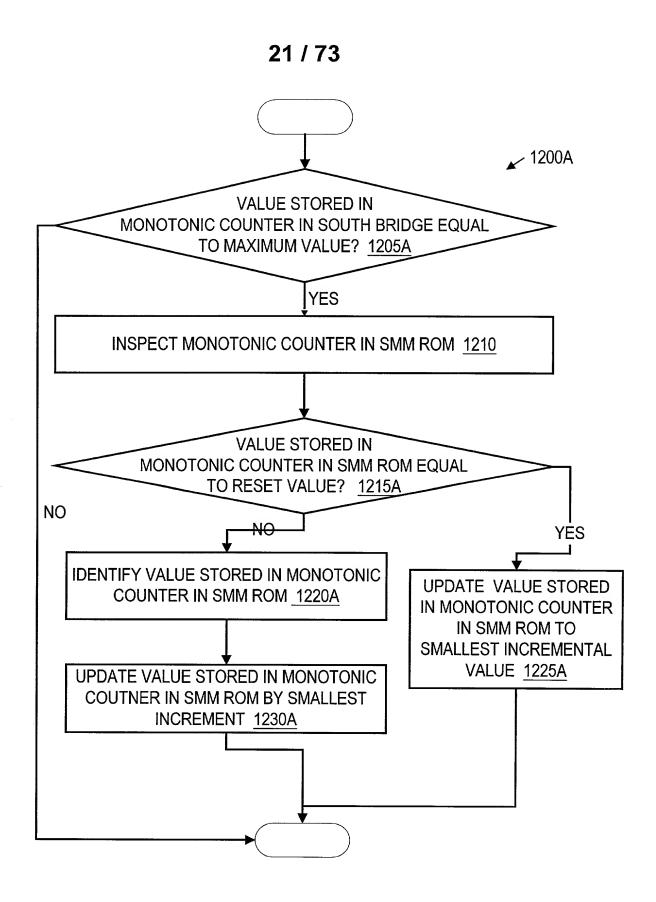


Fig. 11B



**Fig. 12A** 

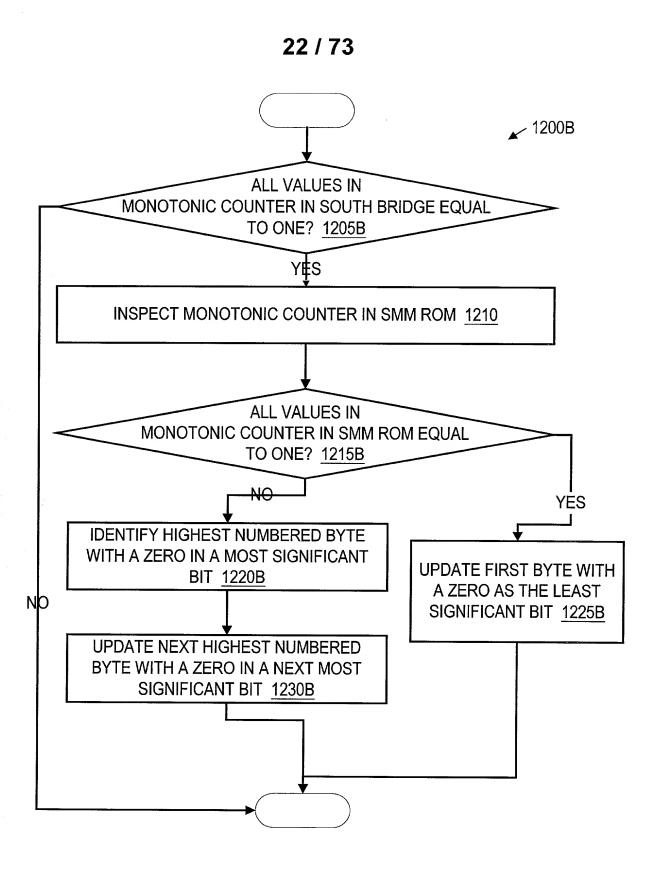
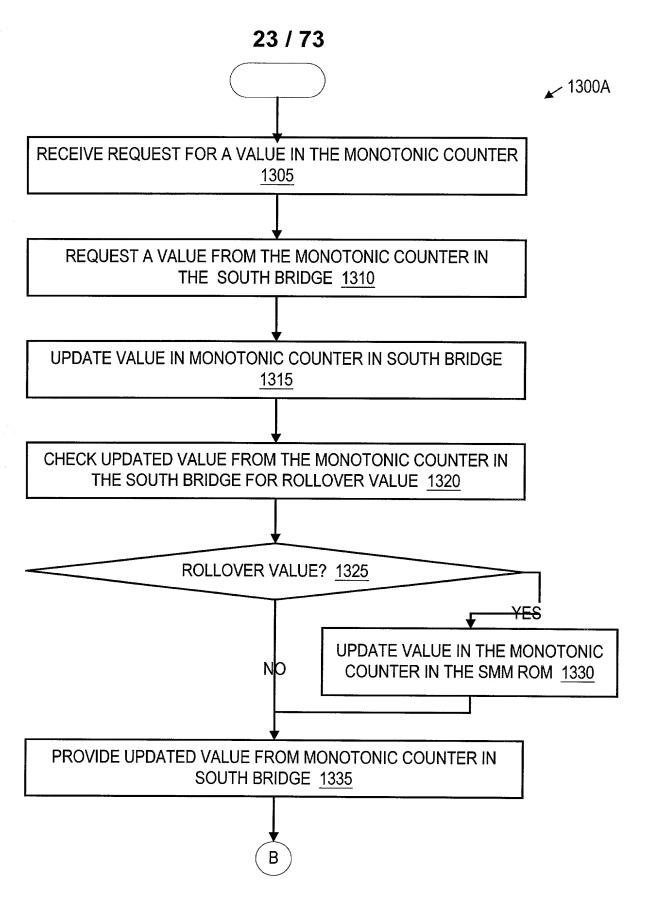


Fig. 12B



**Fig. 13A** 

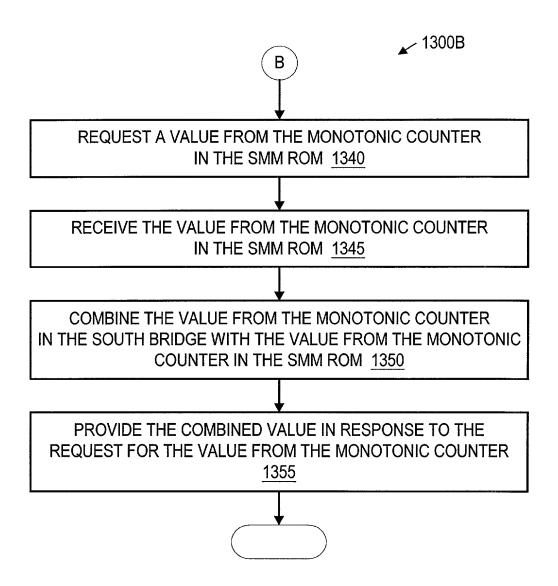


Fig. 13B

25 / 73

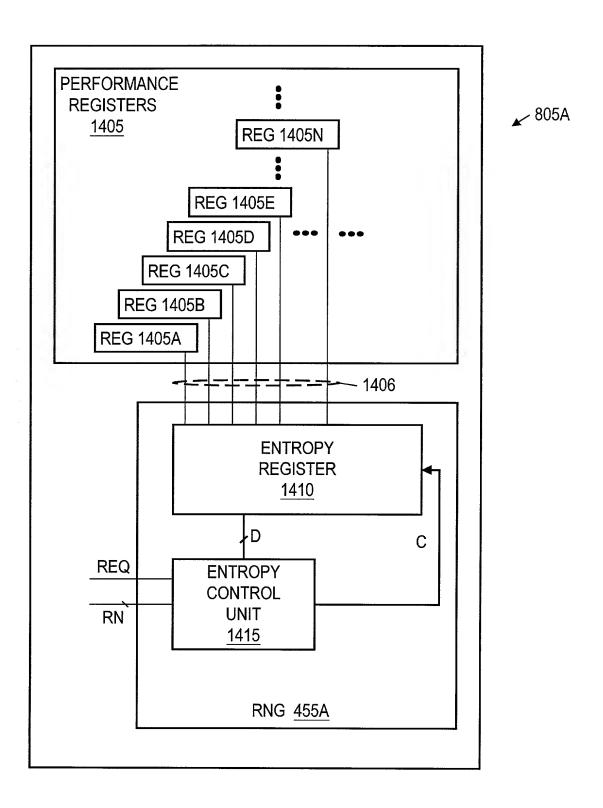


Fig. 14A

26 / 73

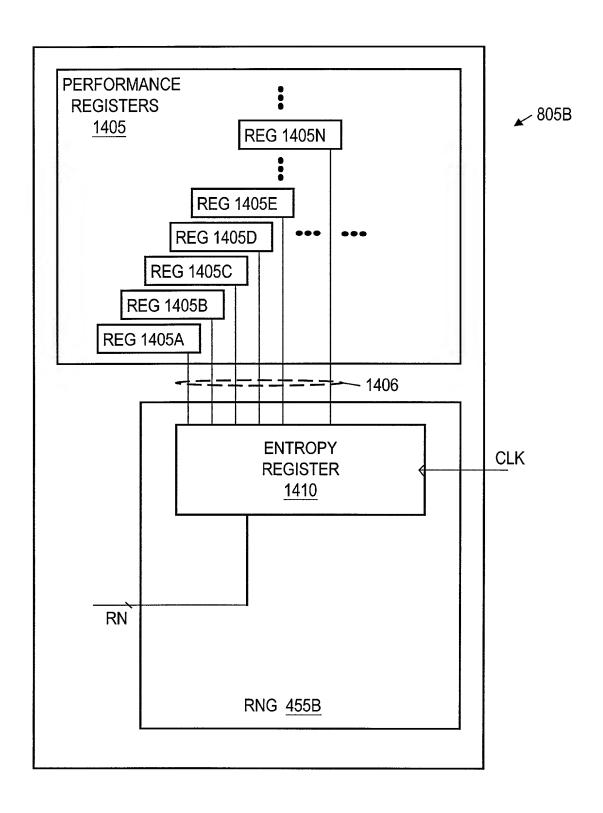


Fig. 14B

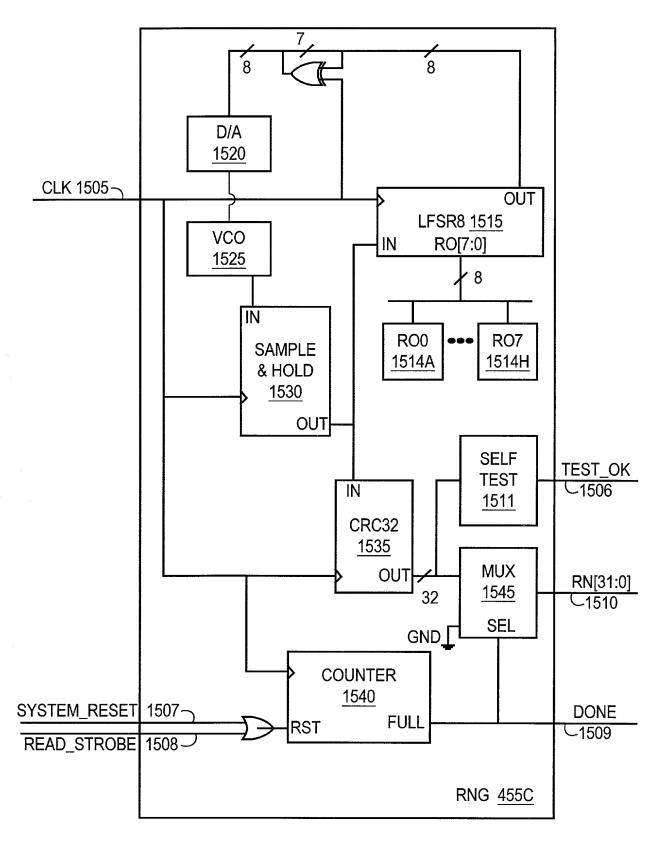
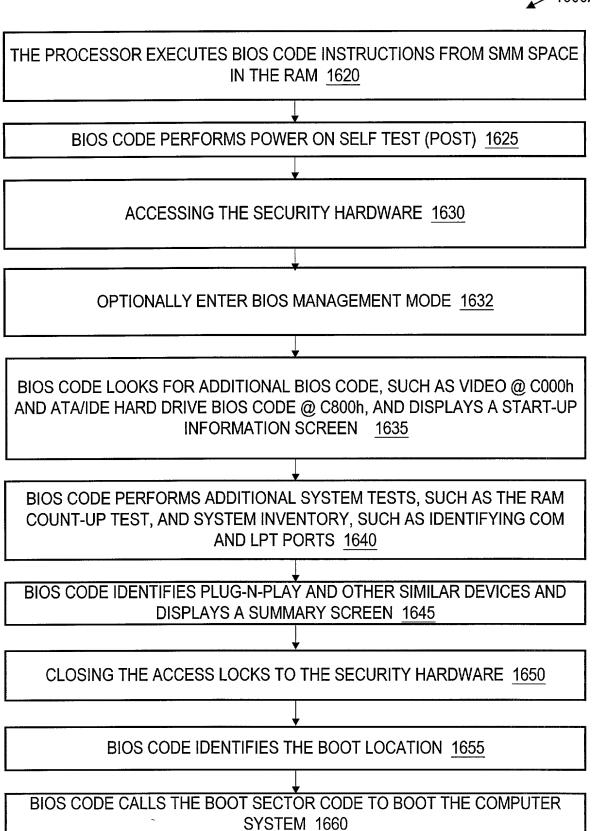


Fig. 15

1600A



**Fig. 16A** 



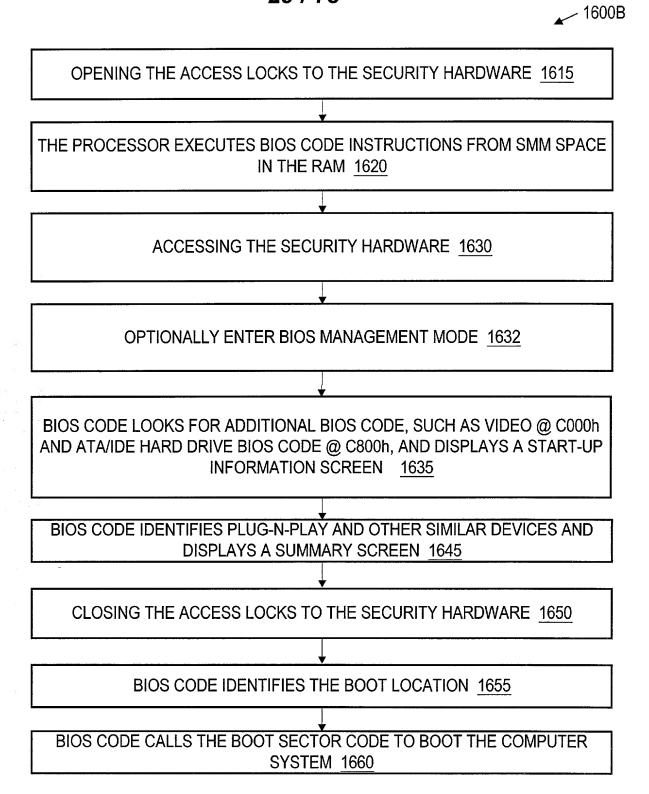


Fig. 16B

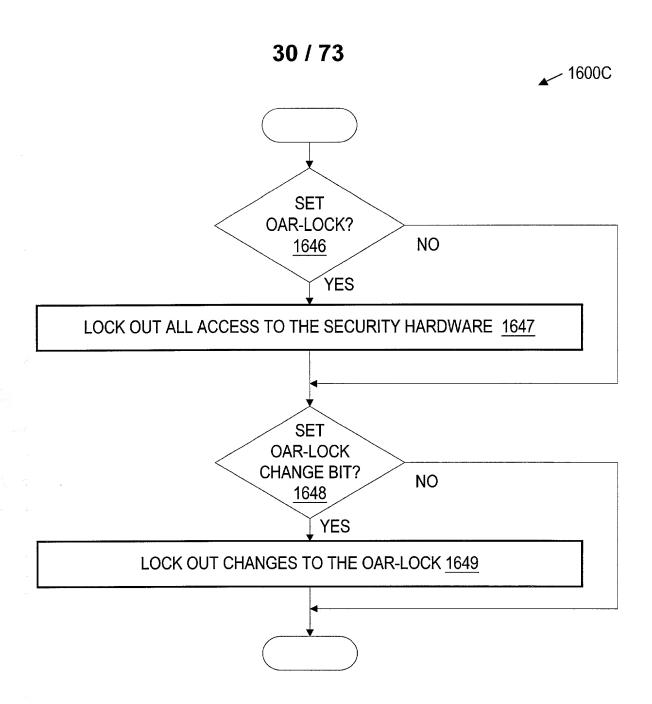
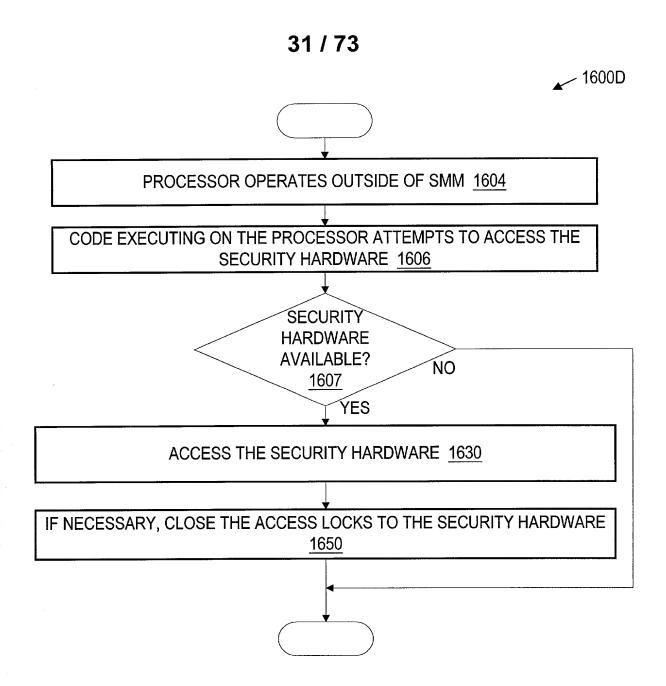


Fig. 16C



**Fig. 16D** 

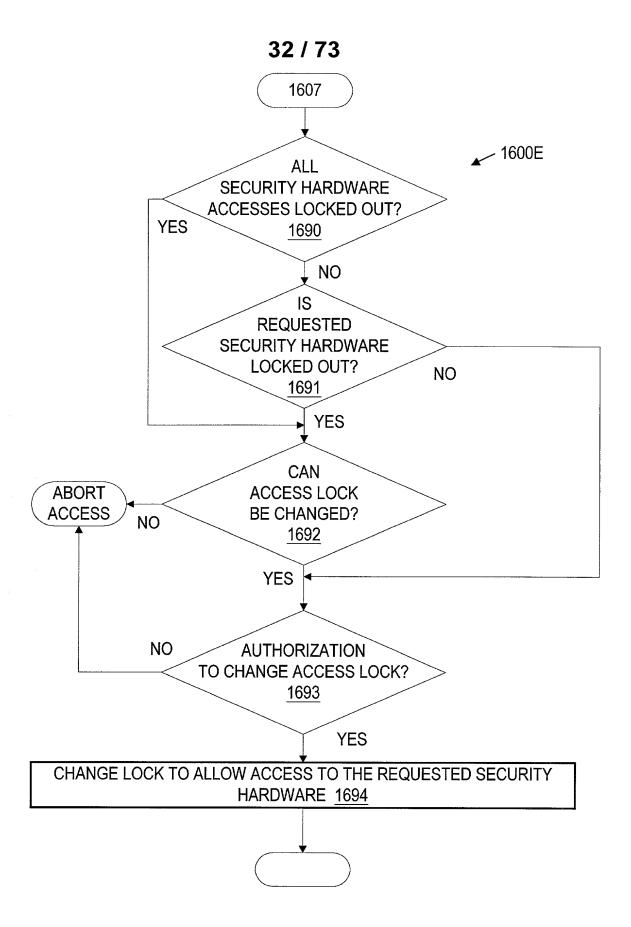


Fig. 16E

THE PROCESSOR LOADS CODE INSTRUCTIONS INTO SMM SPACE IN THE RAM 1605

OPENING THE ACCESS LOCKS TO THE SECURITY HARDWARE 1615

THE PROCESSOR EXECUTES SMM CODE INSTRUCTIONS FROM SMM SPACE IN THE RAM 1620

ACCESSING THE SECURITY HARDWARE 1630

CLOSING THE ACCESS LOCKS TO THE SECURITY HARDWARE 1650

THE PROCESSOR RELOADS THE PREVIOUS STATE AND CONTINUES OPERATING 1665

**Fig. 16F** 

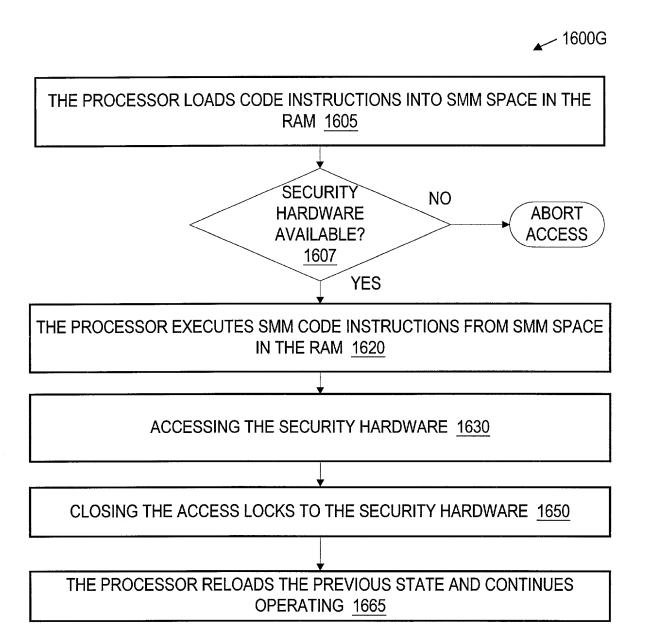
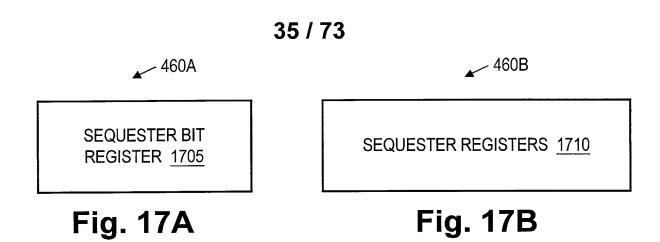
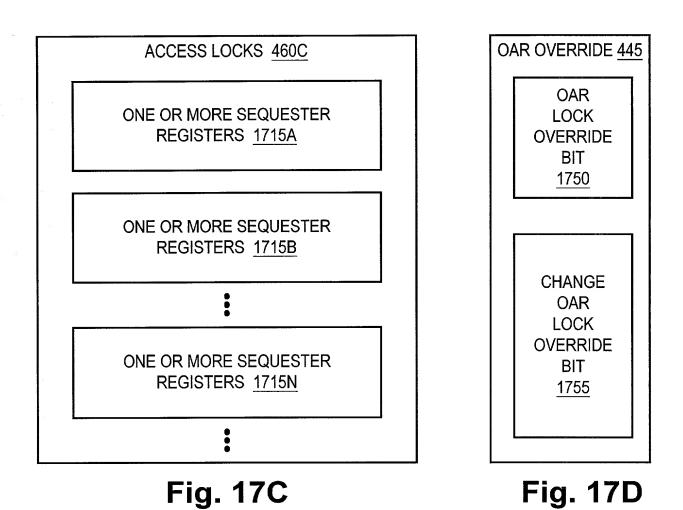
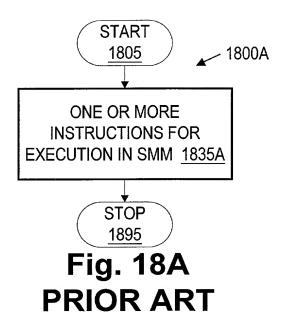


Fig. 16G







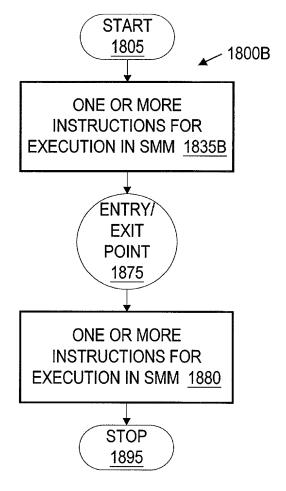


Fig. 18B

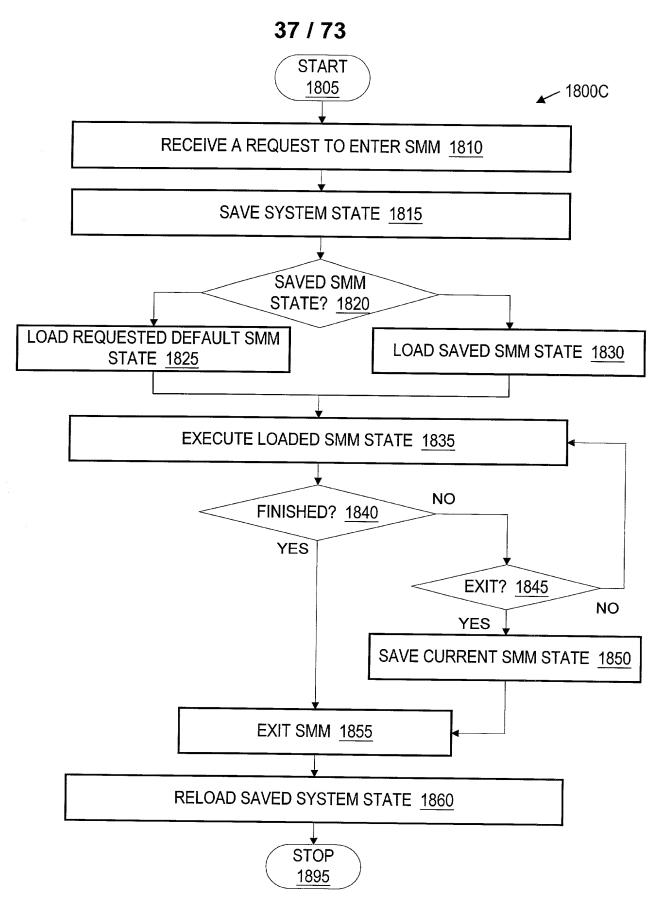


Fig. 18C

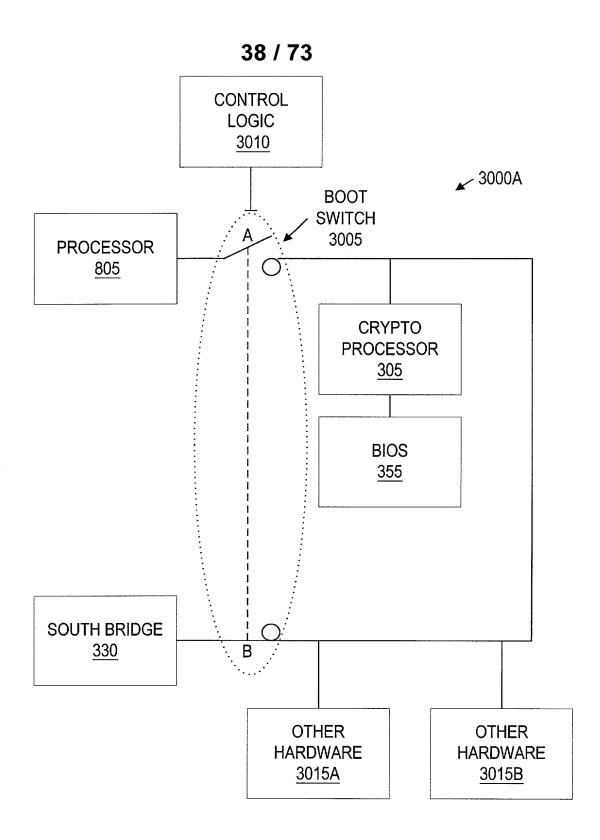
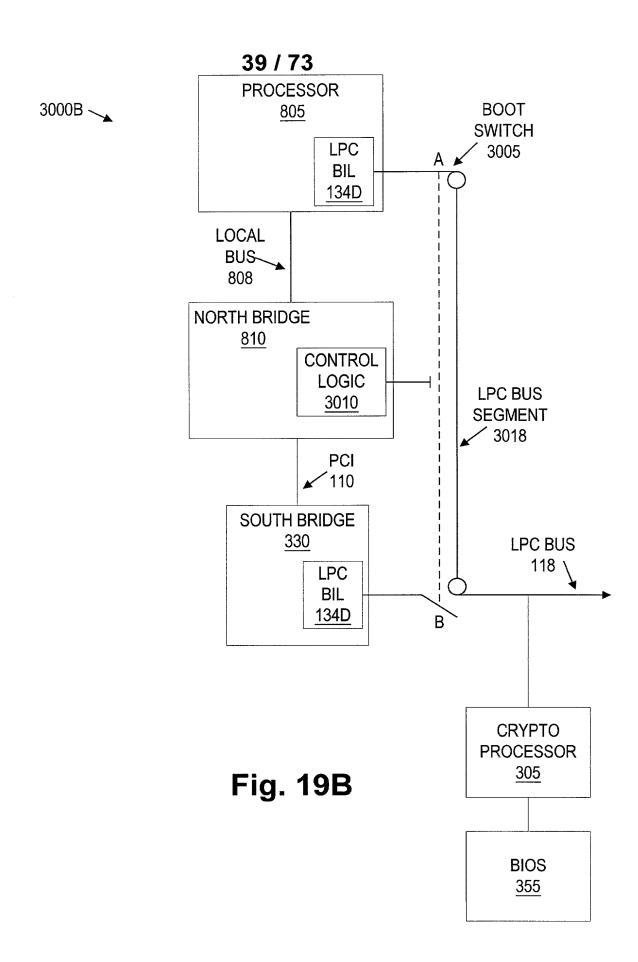


Fig. 19A



40 / 73

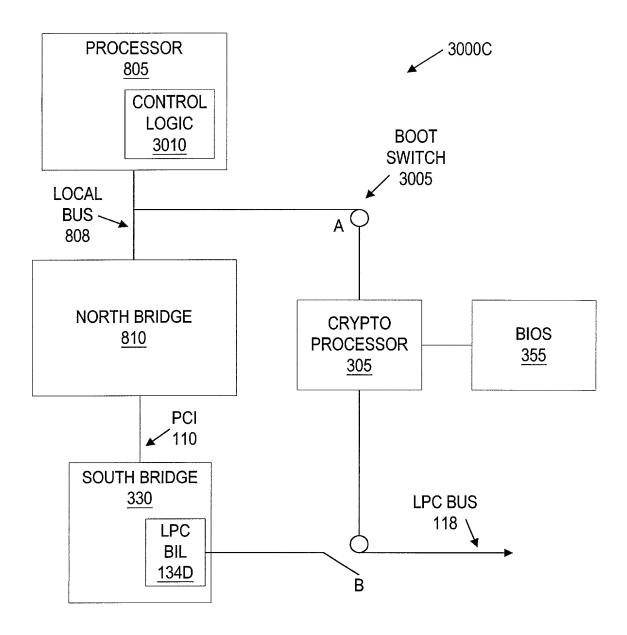


Fig. 19C

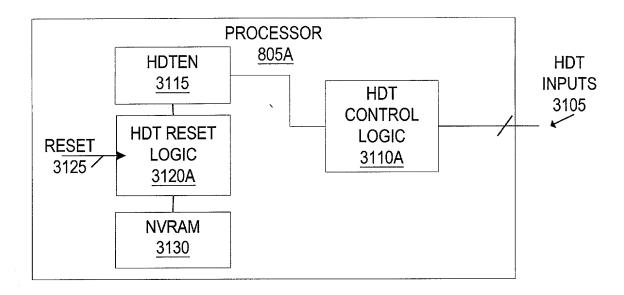
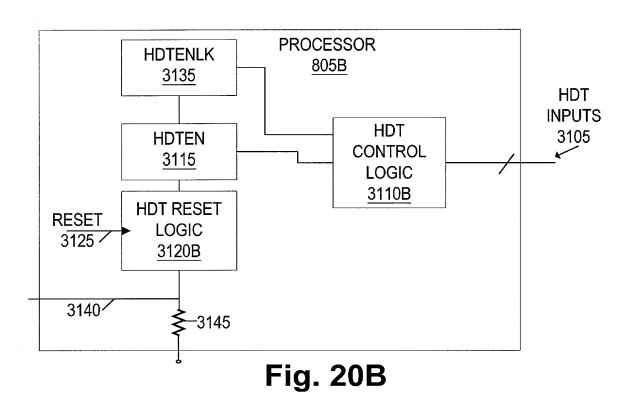
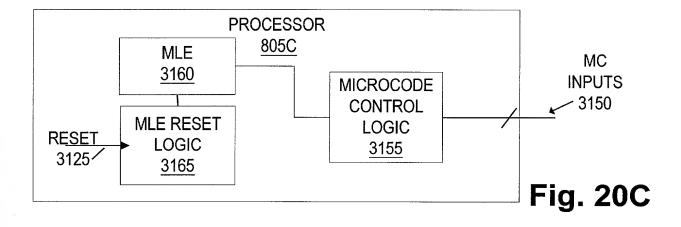
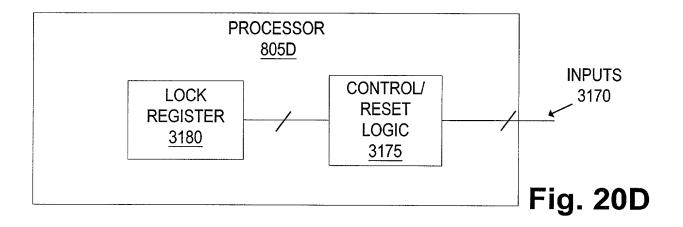


Fig. 20A



42 / 73





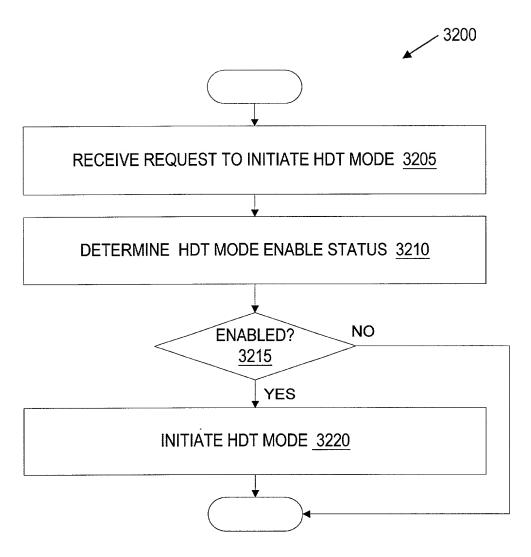


Fig. 21

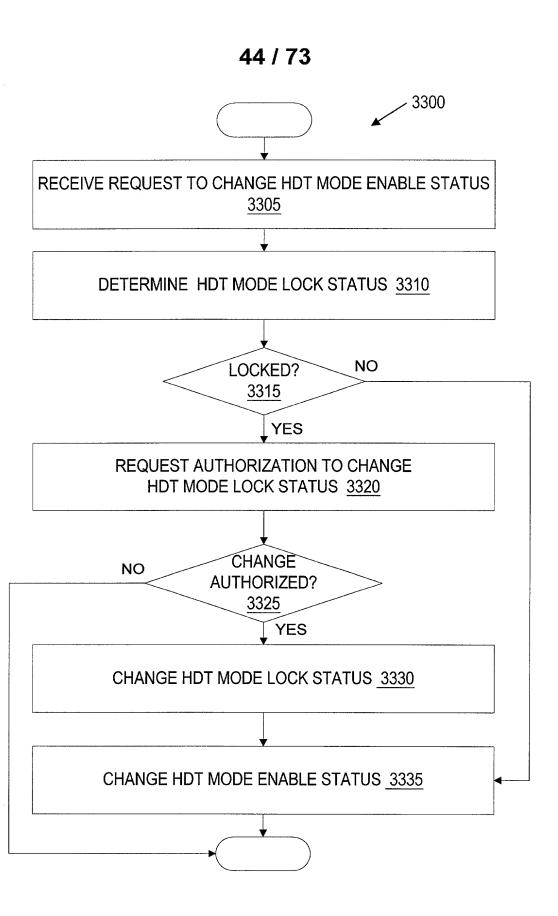


Fig. 22

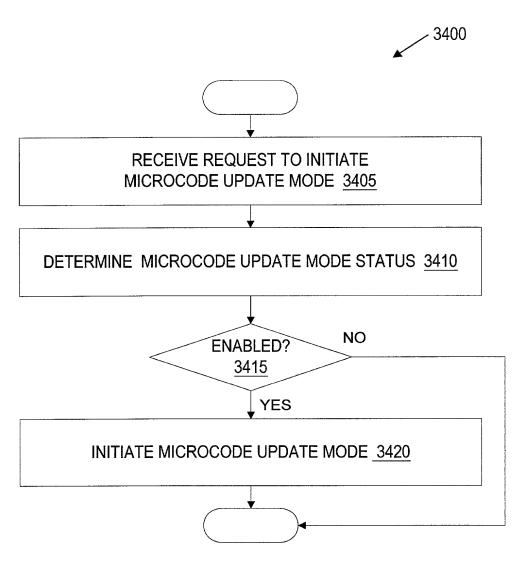


Fig. 23

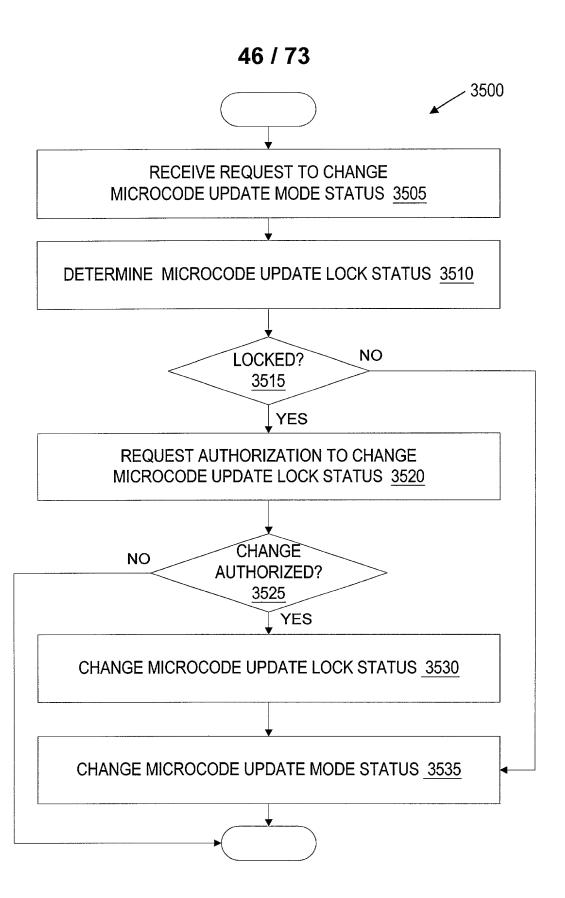
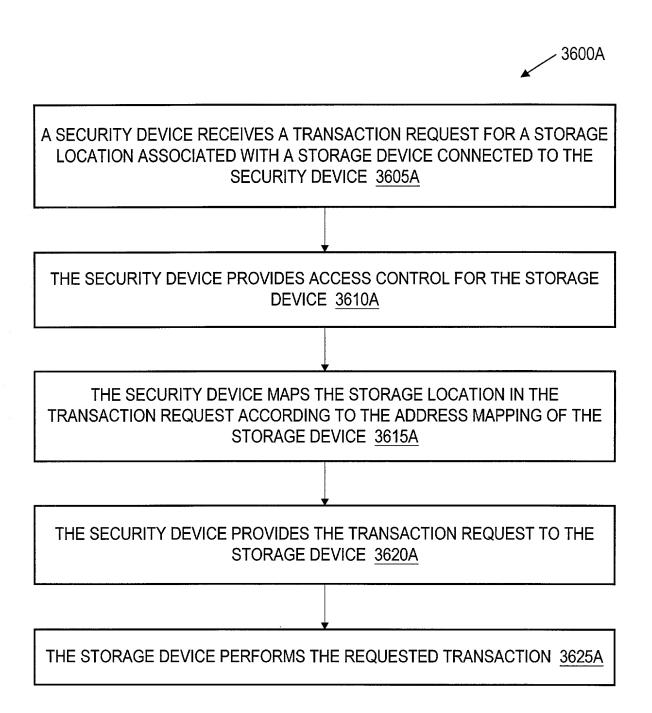


Fig. 24



**Fig. 25A** 

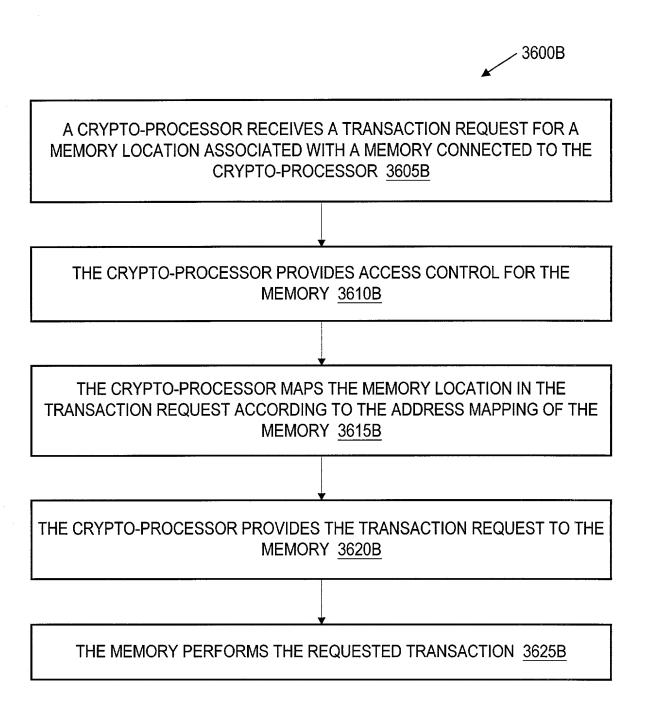


Fig. 25B

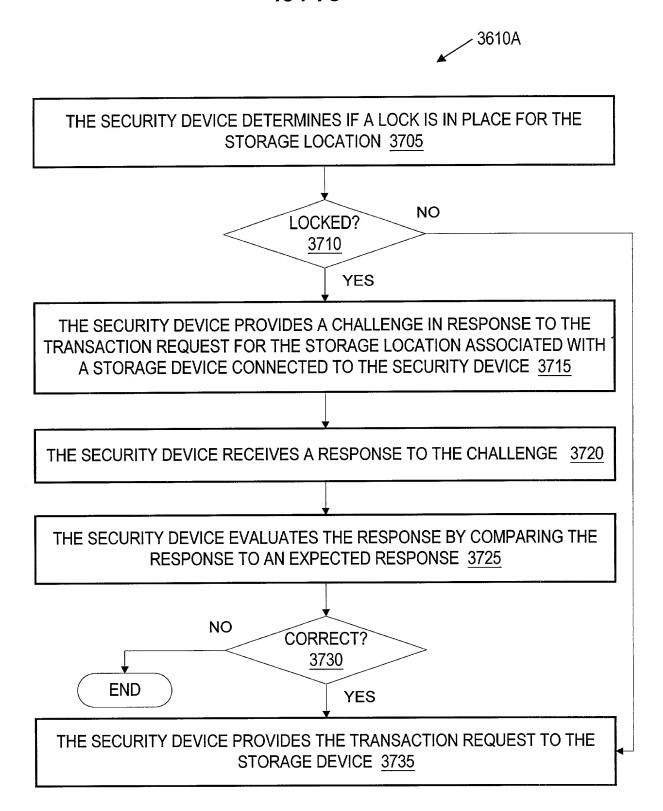


Fig. 26

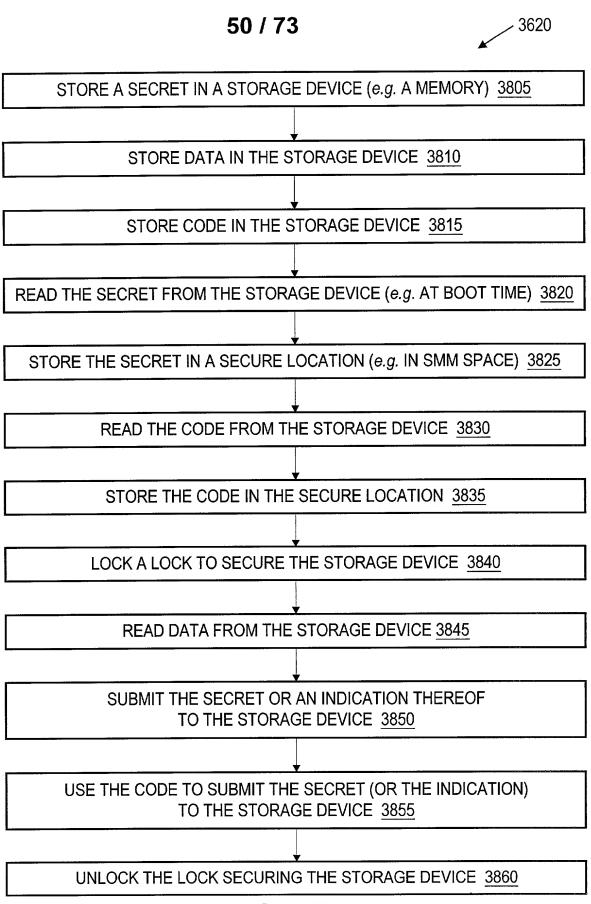


Fig. 27

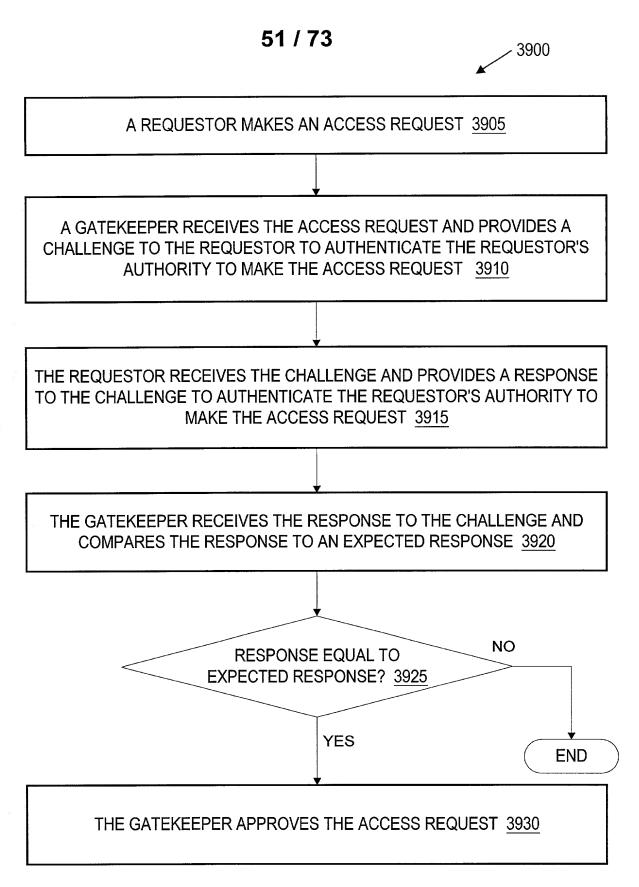
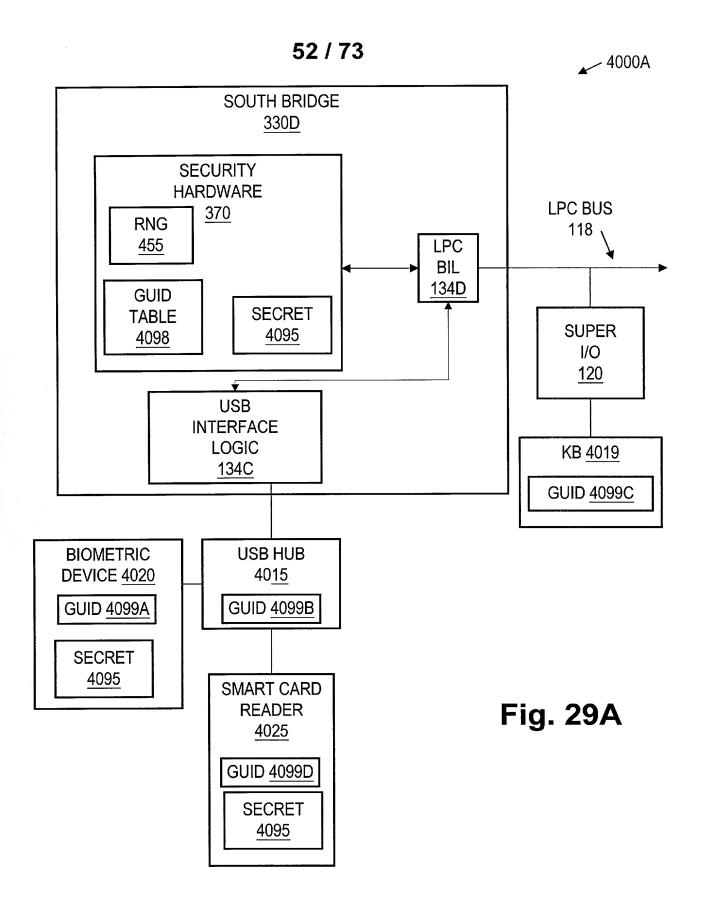


Fig. 28 (Prior Art)



53 / 73

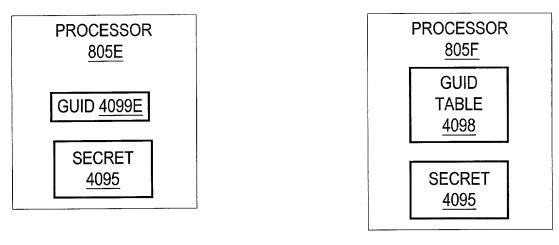
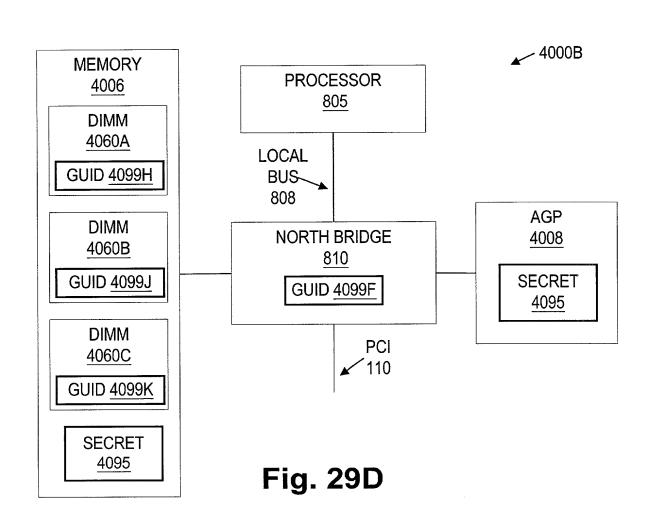
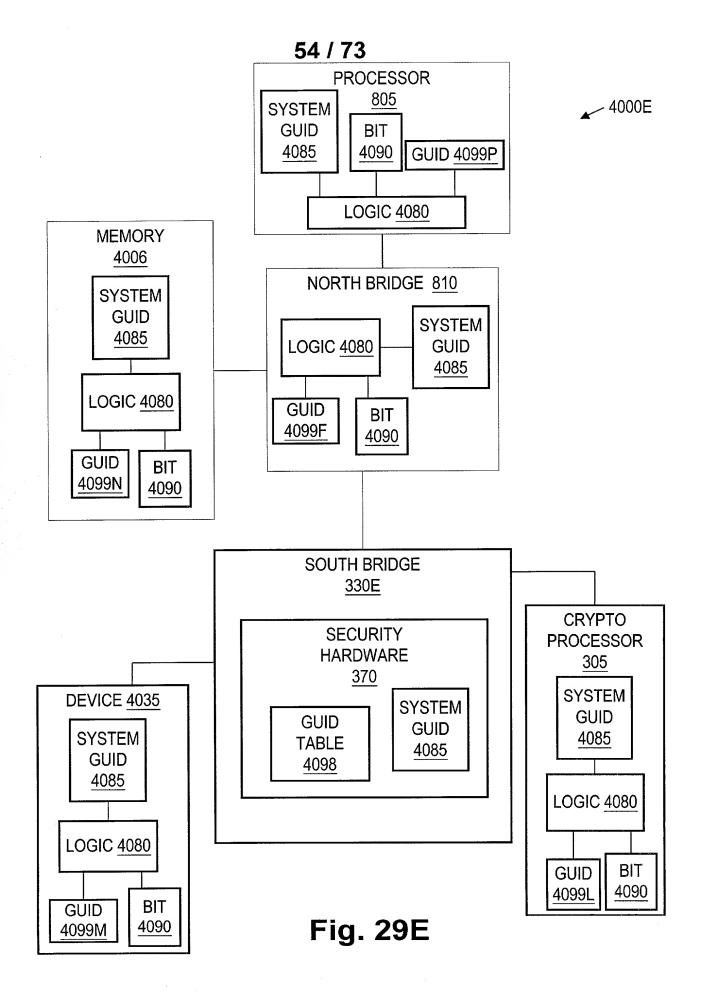


Fig. 29B

Fig. 29C





4100A

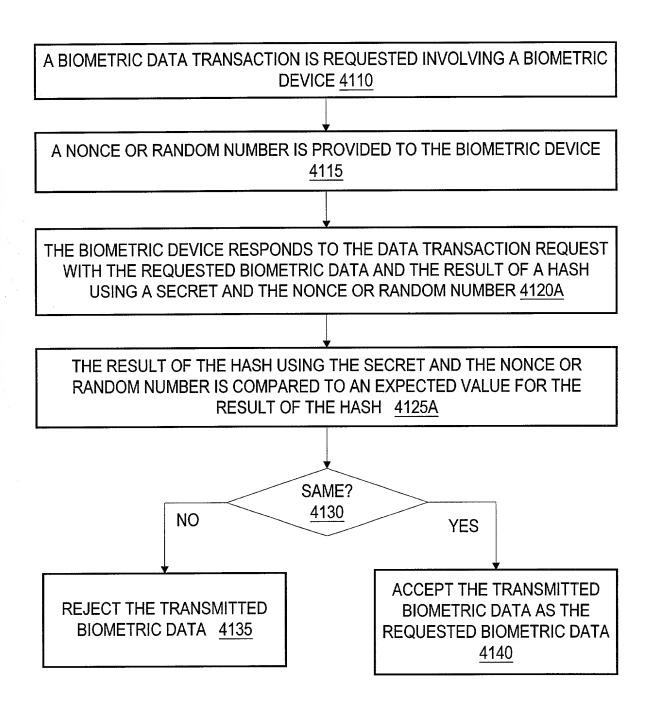
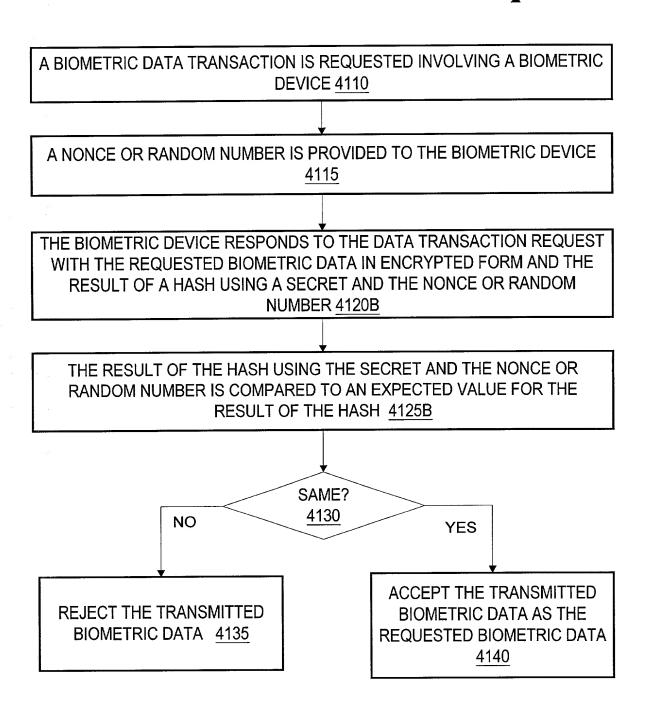


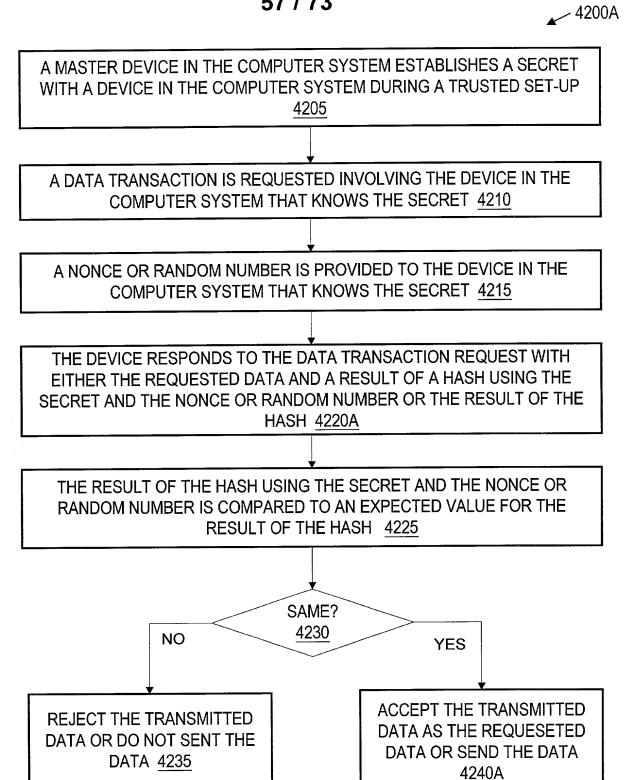
Fig. 30A

4100B

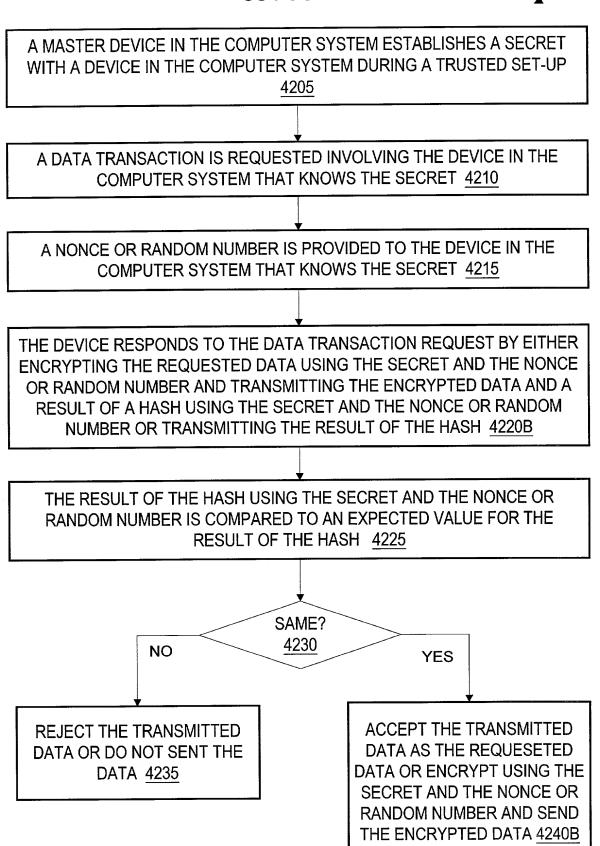


**Fig. 30B** 





**Fig. 31A** 



**Fig. 31B** 



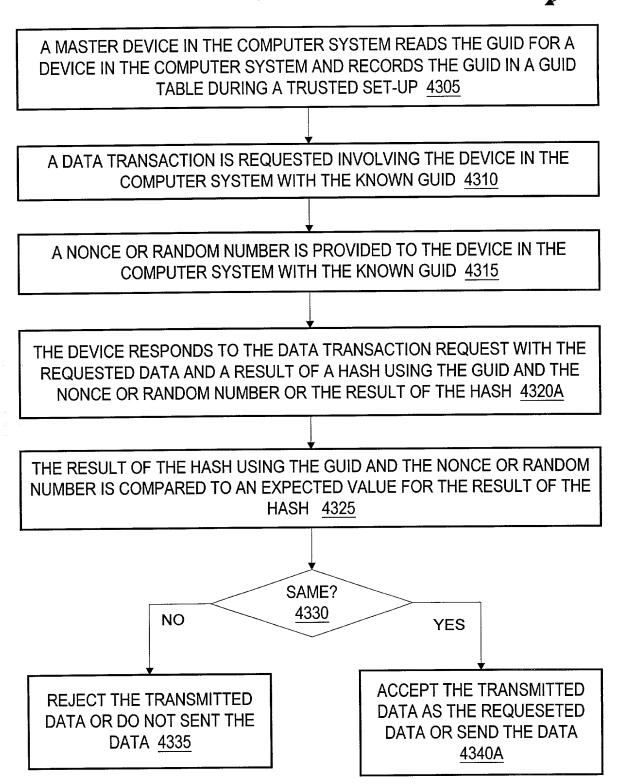


Fig. 32A

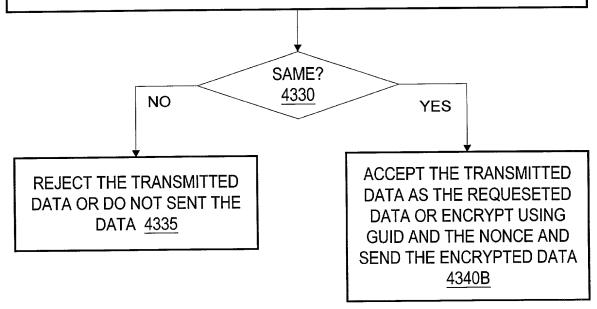
A MASTER DEVICE IN THE COMPUTER SYSTEM READS THE GUID FOR A DEVICE IN THE COMPUTER SYSTEM AND RECORDS THE GUID IN A GUID TABLE DURING A TRUSTED SET-UP 4305

A DATA TRANSACTION IS REQUESTED INVOLVING THE DEVICE IN THE COMPUTER SYSTEM WITH THE KNOWN GUID 4310

A NONCE OR RANDOM NUMBER IS PROVIDED TO THE DEVICE IN THE COMPUTER SYSTEM WITH THE KNOWN GUID 4315

THE DEVICE RESPONDS TO THE DATA TRANSACTION REQUEST BY ENCRYPTING THE REQUESTED DATA USING THE GUID AND THE NONCE OR RANDOM NUMBER AND TRANSMITTING THE ENCRYPTED DATA AND A RESULT OF A HASH USING THE GUID AND THE NONCE OR RANDOM NUMBER OR TRANSMITTING THE RESULT OF THE HASH 4320B

THE RESULT OF THE HASH USING THE GUID AND THE NONCE OR RANDOM NUMBER IS COMPARED TO AN EXPECTED VALUE FOR THE RESULT OF THE HASH 4325



**Fig. 32B** 

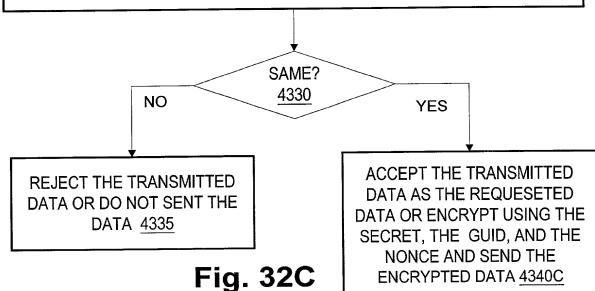
A MASTER DEVICE IN THE COMPUTER SYSTEM READS THE GUID FOR A DEVICE IN THE COMPUTER SYSTEM, RECORDS THE GUID IN A GUID TABLE, AND TRANSMITS A SECRET TO THE DEVICE DURING A TRUSTED SET-UP 4306

A DATA TRANSACTION IS REQUESTED INVOLVING THE DEVICE IN THE COMPUTER SYSTEM WITH THE KNOWN GUID THAT KNOWS THE SECRET 4311

A NONCE OR RANDOM NUMBER IS PROVIDED TO THE DEVICE IN THE COMPUTER SYSTEM WITH THE KNOWN GUID THAT KNOWS THE SECRET 4316

THE DEVICE RESPONDS TO THE DATA TRANSACTION REQUEST BY ENCRYPTING THE REQUESTED DATA USING THE SECRET, THE GUID, AND THE NONCE OR RANDOM NUMBER AND TRANSMITTING THE ENCRYPTED DATA AND A RESULT OF A HASH USING THE SECRET, THE GUID, AND THE NONCE OR RANDOM NUMBER OR TRANSMITTING THE RESULT OF THE HASH 4320C

THE RESULT OF THE HASH USING THE SECRET, THE GUID, AND THE NONCE OR RANDOM NUMBER IS COMPARED TO AN EXPECTED VALUE FOR THE RESULT OF THE HASH 4326



4500

THE DEVICE OR THE MASTER DEVICE INITIATES A REQUEST FOR THE DEVICE TO LEAVE THE COMPUTER SYSTEM 4505

THE DEVICE AND THE MASTER DEVICE AUTHENTICATE EACH OTHER USING THE GUID AND/OR THE SYSTEM GUID IN RESPONSE TO THE REQUEST FOR THE DEVICE TO LEAVE THE COMPUTER SYSTEM 4510

THE DEVICE RESETS THE INTRODUCED BIT IN RESPONSE TO THE DEVICE AND THE MASTER DEVICE SUCCESSFULLY AUTHENTICATING EACH OTHER  $\frac{4515}{}$ 

Fig. 34

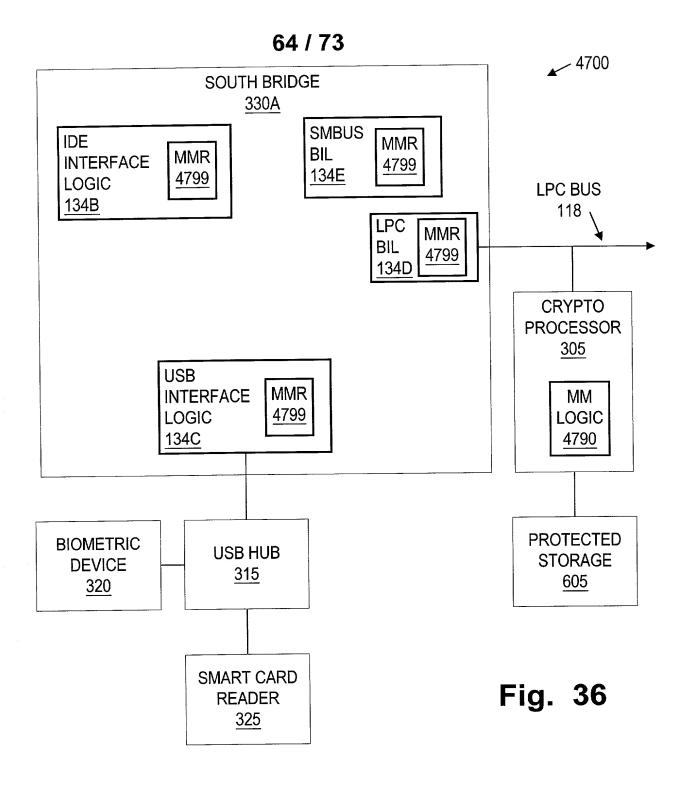
4600

THE DEVICE RECEIVING A COMMAND FOR THE DEVICE TO LEAVE THE COMPUTER SYSTEM  $\underline{4605}$ 

THE DEVICE RECEIVING A MAINTENANCE KEY THAT SUCCESSFULLY AUTHENTICATES 4610

THE DEVICE RESETS THE INTRODUCED BIT IN RESPONSE TO THE DEVICE RECEIVING THE MAINTENANCE KEY THAT SUCCESSFULLY AUTHENTICATES 4615

Fig. 35



4800

TRANSMIT A MASTER MODE SIGNAL TO BUS INTERFACE LOGIC CONNECTED BETWEEN MASTER MODE LOGIC AND A DATA INPUT DEVICE, WHERE THE BUS INTERFACE LOGIC INCLUDES A MASTER MODE REGISTER 4805

SET A MASTER MODE BIT IN THE MASTER MODE REGISTER(S) TO ESTABLISH SECURE TRANSMISSION CHANNEL BETWEEN THE MASTER MODE LOGIC AND THE DATA INPUT DEVICE OUTSIDE THE OPERATING SYSTEM OF THE COMPUTER SYSTEM 4810

THE MASTER MODE LOGIC AND THE DATA INPUT DEVICE EXCHANGE DATA OUTSIDE THE OPERATING SYSTEM OF THE COMPUTER SYSTEM THROUGH THE BUS INTERFACE LOGIC(S) THAT INCLUDE THE MASTER MODE REGISTER 4815

THE MASTER MODE LOGIC FLUSHES THE BUFFERS OF THE BUS INTERFACE LOGIC(S) THAT INCLUDE THE MASTER MODE REGISTER AFTER CONCLUDING THE DATA TRANSMISSIONS 4820

THE MASTER MODE LOGIC SIGNALS THE BUS INTERFACE LOGIC(S) TO UNSET THE MASER MODE BITS AFTER FLUSHING THE BUFFERS OF THE BUS INTERFACE LOGIC(S) THAT INCLUDE THE MASTER MODE REGISTER 4825

Fig. 37

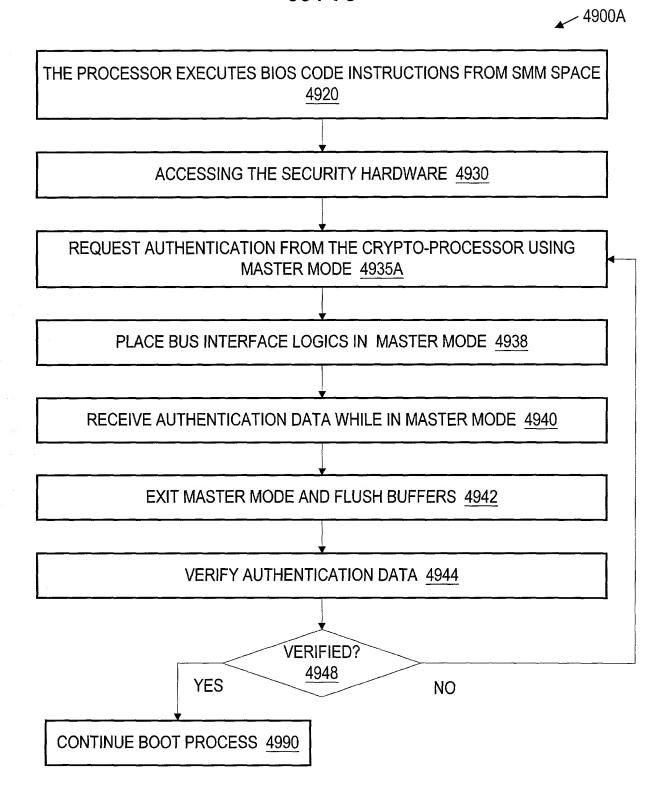
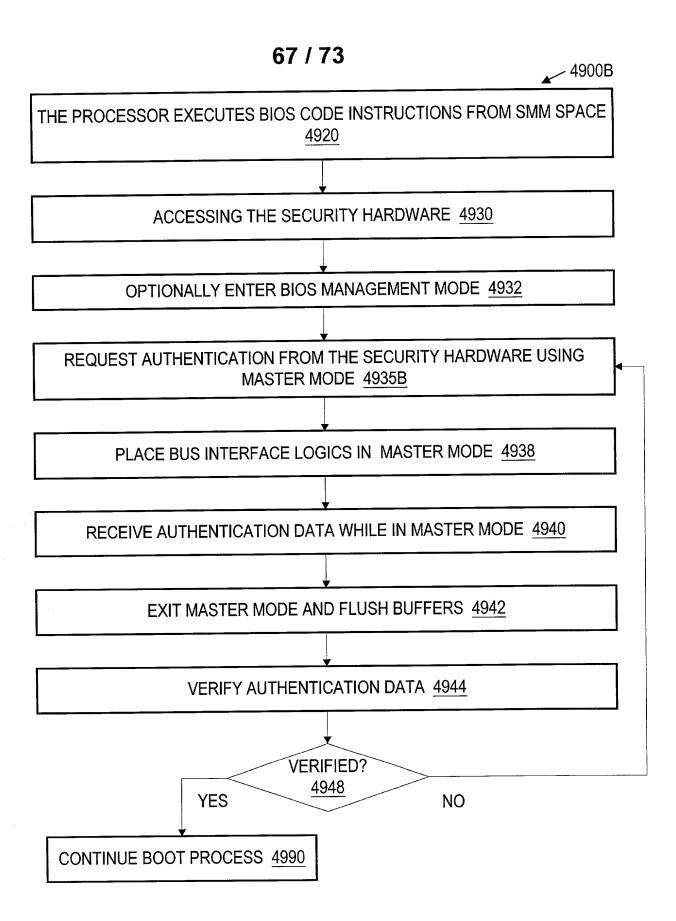


Fig. 38A



**Fig. 38B** 

68 / 73

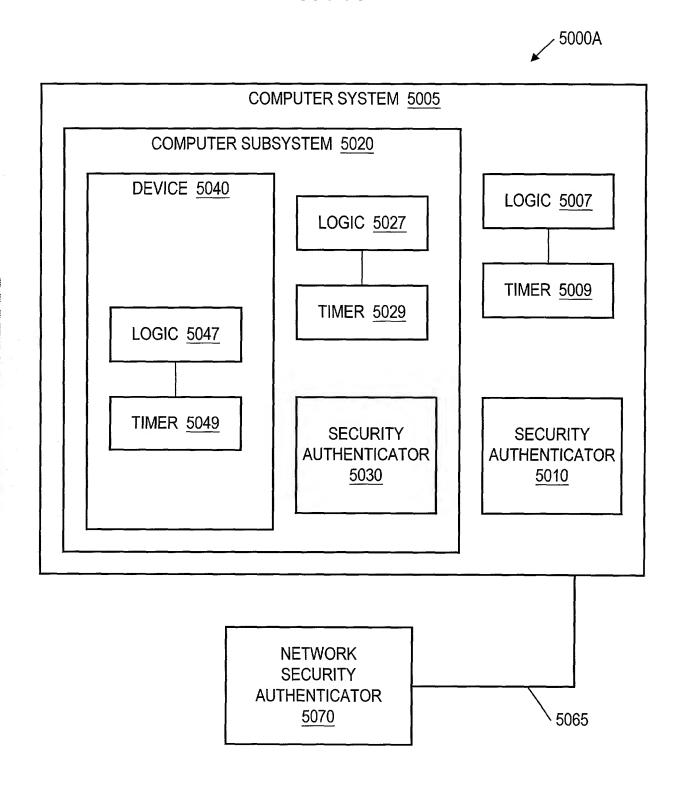
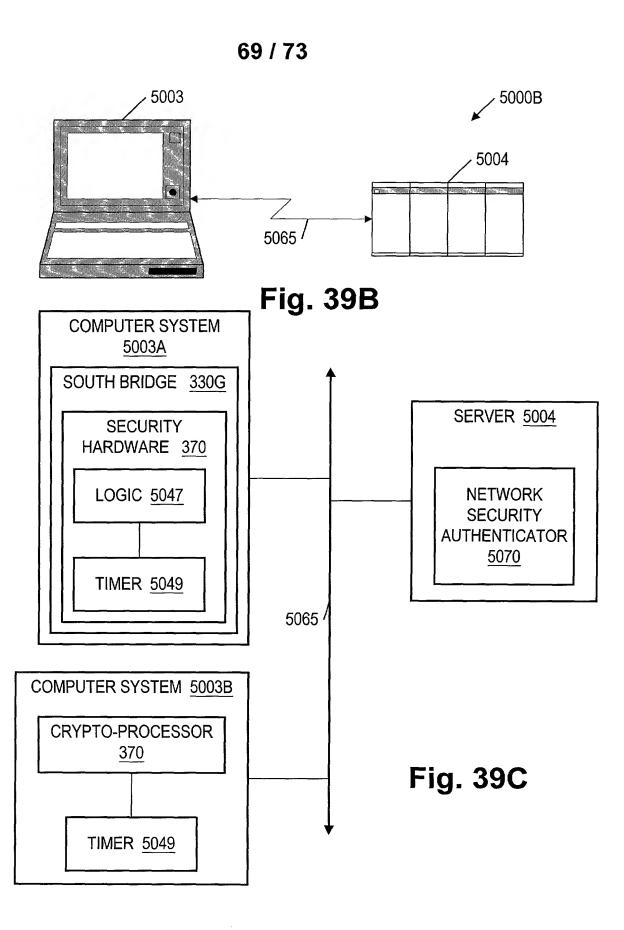
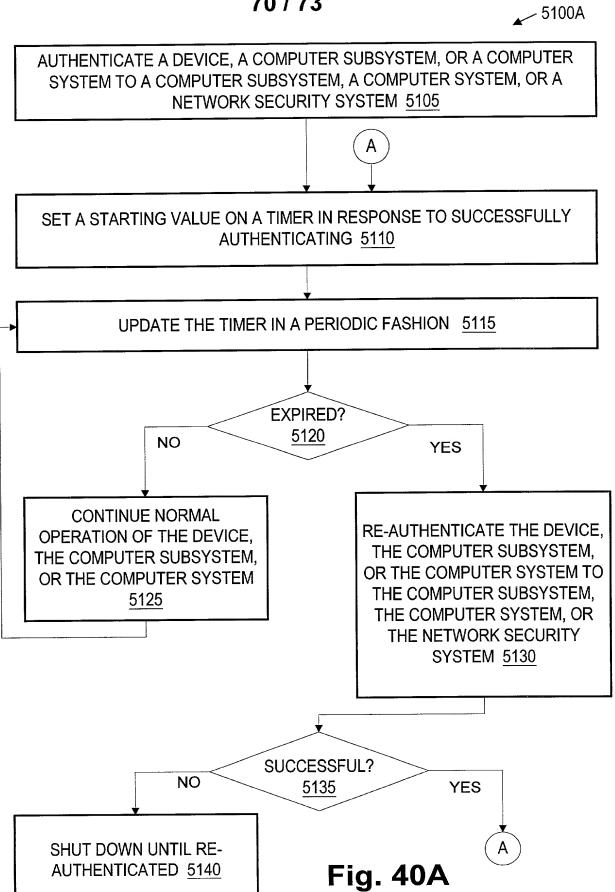
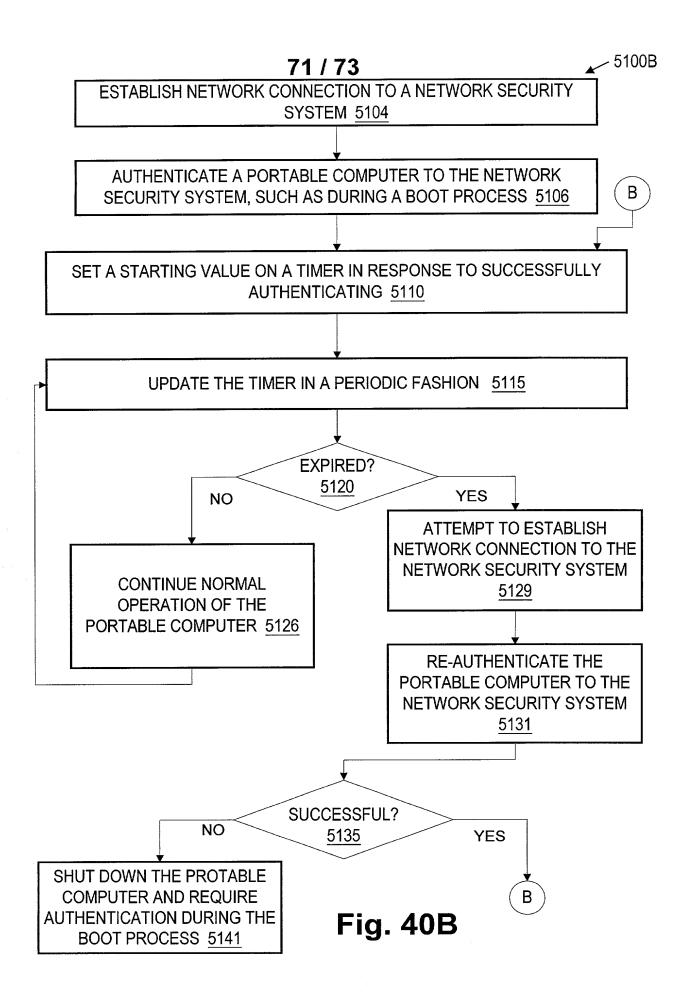


Fig. 39A







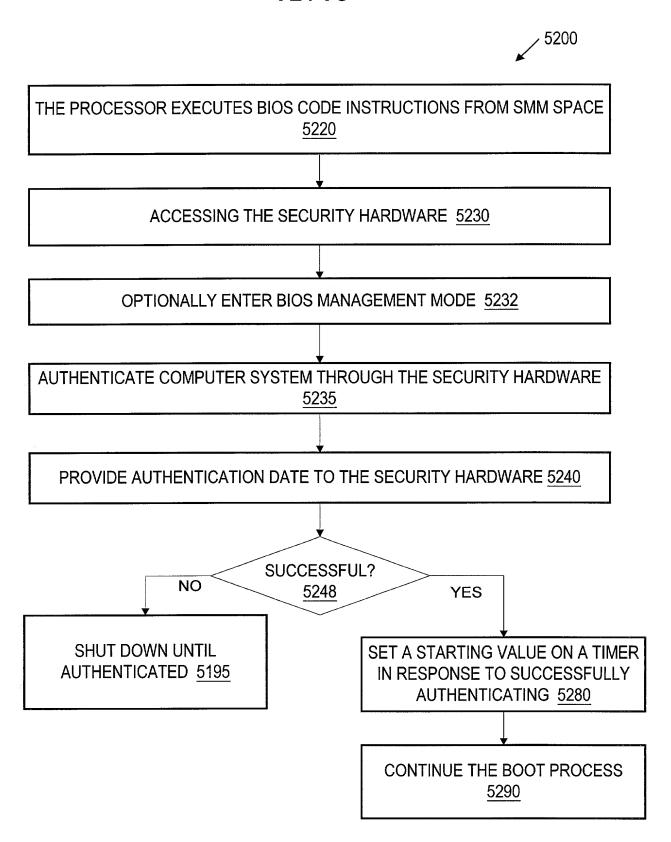


Fig. 41

73 / 73

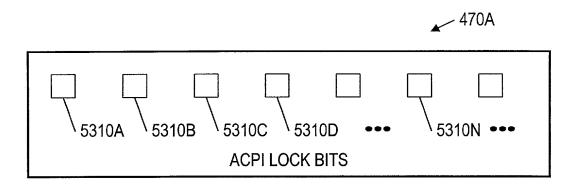


Fig. 42A

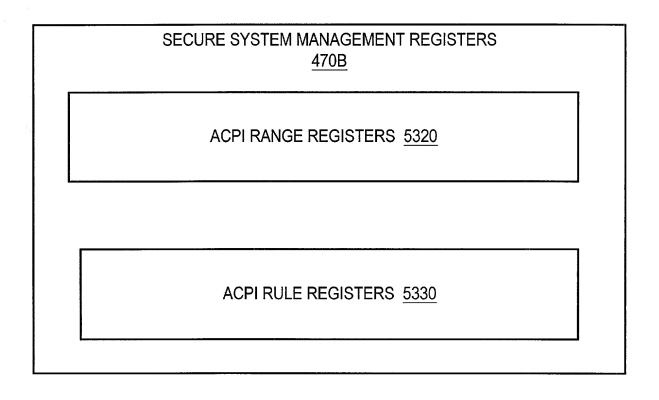


Fig. 42B